NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

ITERATIVE SYSTEM MODELING
USING
MULTIGRID TECHNIQUES

by

Dean A. Richter

December, 1991

Thesis Advisor: Murali Tummala

Approved for public release; distribution is unlimited

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED	1b RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3 DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5 MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (*If applicable*) EC	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000	7b ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (*If applicable*)	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (*City, State, and ZIP Code*)	10 SOURCE OF FUNDING NUMBERS Program Element No / Project No / Task No / Work Unit Accession Number

11. TITLE (*Include Security Classification*)
ITERATIVE SYSTEM MODELING USING MULTIGRID METHODS

12. PERSONAL AUTHOR(S) RICHTER, Dean Anthony

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (*year, month, day*)	15. PAGE COUNT
Master's Thesis	From To	1991 December	135

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES			18. SUBJECT TERMS (*continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUBGROUP	finite impulse response modeling; autoregressive modeling; infinite impulse response modeling; 2-D autoregressive modeling; spectral estimation; iterative methods; Toeplitz; multigrid techniques

19 ABSTRACT (*continue on reverse if necessary and identify by block number*)

One and two-dimensional system identification and modeling algorithms utilizing multigrid techniques are presented. Finite impulse response (FIR), autoregressive (AR), infinite impulse response (IIR), and 2-D block matrix iterative system modeling algorithms are enhanced and made more efficient using the multigrid methods. The convergence performance of these algorithms is improved with the multigrid techniques. The reduction in the number of iterations required to converge to a solution is realized by forcing the low frequency error components to appear to be at a higher frequency by transferring to a coarser sampling period. Performance comparisons are presented for FIR, AR, IIR, and 2-D block matrix modeling simulations with and without the multigrid techniques employed.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL TUMMALA, Murali	22b TELEPHONE (*Include Area code*) (408) 646-2645	22c. OFFICE SYMBOL EC/Tu

DD FORM 1473, 84 MAR 83 APR edition may be used until exhausted
All other editions are obsolete

Approved for public release; distribution is unlimited.

Iterative System Modeling Using Multigrid Techniques

by

Dean A. Richter
Lieutenant, United States Navy
B.S.M.E., University of Missouri - Rolla , 1983

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1991

ABSTRACT

One and two-dimensional system identification and modeling algorithms utilizing multigrid techniques are presented. Finite impulse response (FIR), autoregressive (AR), infinite impulse response (IIR), and 2-D block matrix iterative system modeling algorithms are enhanced and made more efficient using the multigrid methods. The convergence performance of these algorithms is improved with the multigrid techniques. The reduction in the number of iterations required to converge to a solution is realized by forcing the low frequency error components to appear to be at a higher frequency by transferring to a coarser sampling period. Performance comparisons are presented for FIR, AR, IIR, and 2-D block matrix modeling simulations with and without the multigrid techniques employed.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENT

The successful completion of this thesis could not have been possible without the undying support of many individuals. I must express my love and devotion to my dear wife, Toni, and our children, Shane and Brittney, who provided me with the unwavering love, inspiration, motivation, and support that I needed throughout my entire graduate education. I must also thank and express gratitude to my thesis advisor, Dr. Murali Tummala, whose gifted talents as a teacher drew me to him and whose wisdom and intelligence kept me on track during the entire thesis process. I also am indebted to my second reader, Dr. Charles W. Therrien and to all of the faculty and staff of the Electrical and Computer Engineering Department whose knowledge, inspiration, and guidance I am eternally grateful for.

I. INTRODUCTION

A. PARAMETER ESTIMATION METHODS

In most system modeling algorithms, a system of linear equations of the form $Ra=r$ is formed where R is an autocorrelation matrix, r is a cross-correlation vector, and a is the vector of system parameters or coefficients. The solution of this set of linear equations can be arrived at using many methods. It can be solved directly by matrix inversion [Ref. 1], but this is usually undesirable since matrix inversion is computationally expensive. Iterative methods, using a Toeplitz approximation algorithm [Refs. 2,3,4,5,6], provide an alternative to the direct matrix inversion solution approach. The iterative Toeplitz approximation method has proven to be an effective means to estimate parameters of fixed data arrays without having to invert the correlation matrix.

Examining the convergence properties of some of the basic iterative schemes, as well as the Toeplitz approximation methods, shows that most of these schemes work very well for the first several iterations [Ref. 2]. Eventually, however, the convergence slows and the entire scheme appears to stall [Ref. 7]. The rapid decrease in error during the early iterations is due to the efficient elimination of the high frequency components of the error. Once the high frequency components have been removed, the iteration is much less effective in reducing the remaining low frequency components. This smoothing property, i.e. eliminating the high frequency components and leaving the low frequency components of the error, is common to many iterative methods. This is, of

course, a serious limitation of these methods. These limitations can be overcome by using the multigrid technique [Ref. 7]. That is the topic of this research.

B. THESIS OUTLINE

The following describes the organization of the remainder of this thesis. Chapter II introduces the reader to the theory of the multigrid technique and the Toeplitz approximation iterative algorithm. It presents the multigrid V-cycle (*V-cycle*), the full multigrid V-cycle (*FMV-cycle*), and draws theoretical performance comparisons between the straight iterative methods and the *V-cycle* and *FMV-cycle* methods. Chapter III develops the basic system modeling algorithms of FIR, AR, IIR, and 2-D block matrix systems are developed along with the 2-D AR Spectral Estimation problem. Particular attention is paid to the integration of the multigrid techniques into these modeling algorithms. Chapter III serves as the foundation for Chapter IV. Chapter IV presents the results of the computer simulations of the algorithms developed in Chapter III. The main purpose of this chapter is to show the increased convergence performance that the multigrid technique, especially the *FMV-cycle* method, offers over the straight iterative methods. Chapter V presents conclusions and suggestions for future work. The appendices provide a discussion of each modeling problem along with more detailed simulation results.

II. ELEMENTS OF THE MULTIGRID TECHNIQUE

A. THE THEORY OF MULTIGRID

Many standard iterative methods possess the smoothing property as described in Chapter I. These methods are thus very effective at eliminating the high frequency or oscillatory components of the error, while leaving the low frequency or smooth components relatively unchanged. One way to enhance any iterative method is to start with a good initial guess. A technique for obtaining an improved initial guess is to perform some preliminary iterations on a coarse grid[1] and then use the resulting approximation as the initial guess on the original fine grid. This idea is computationally less expensive since there are fewer unknowns to be updated. The coarser grid will also have a marginally improved convergence rate. Thus, when the iteration begins to stall, a transfer to a coarser grid will help the iteration to proceed more effectively. [Ref. 7]

In the system of equations $Ra = r$, if $\hat{a}$ is an approximation to the exact solution a, then the error $e = a - \hat{a}$ satisfies

$$Re = d = r - R\hat{a} \; . \qquad (2.1)$$

This is called the residual equation and suggests that iteration can be performed directly on the error itself by using the residual, d.

---

[1]The grid or set of sampling periods, denoted by S^t, consists of N subintervals where $t=1/N$ is the constant width of the subintervals otherwise referred to as the sampling period (See Figure 2.1).

In order to discuss multigrid procedures further, a discussion of the intergrid transfer operators that convert between fine and coarse sampling periods is in order. The two intergrid transfer operators are the *linear* and the *full weighting* operator. The linear operator converts vectors from a coarse sampling period to a fine sampling period (i.e. $\Re^{(N/2-1)}$ to $\Re^{(N-1)}$ space). For the case of N=8 it is implemented as

$$I_{2t}^{t}\hat{a}^{2t} = \frac{1}{2}\begin{bmatrix}1&0&0\\2&0&0\\1&1&0\\0&2&0\\0&1&1\\0&0&2\\0&0&1\end{bmatrix}\begin{bmatrix}\hat{a}_1\\\hat{a}_2\\\hat{a}_3\end{bmatrix}_{2t} = \begin{bmatrix}\hat{a}_1\\\hat{a}_2\\\hat{a}_3\\\hat{a}_4\\\hat{a}_5\\\hat{a}_6\\\hat{a}_7\end{bmatrix}_{t} = \hat{a}^{t}. \qquad (2.2)$$

Note the superscript 't' denotes the fine grid interval or the original sampling period while the superscript '2t' denotes the coarse grid or the coarse sampling period. The set of all fine sampling periods (fine grid) is denoted by S^t while sets of coarser sampling periods (coarse grids) are denoted by S^{2t} and S^{4t} (see Figure 2.1).

The full weighting operator converts vectors from fine sampling periods to coarse sampling periods. It transfers vectors from $\Re^{(N-1)}$ to $\Re^{(N/2-1)}$ space and for N=8 has the form

$$I_t^{2t}\hat{a}^t = \frac{1}{4}\begin{bmatrix}1&2&1&0&0&0&0\\0&0&1&2&1&0&0\\0&0&0&0&1&2&1\end{bmatrix}\begin{bmatrix}\hat{a}_1\\\hat{a}_2\\\hat{a}_3\\\hat{a}_4\\\hat{a}_5\\\hat{a}_6\\\hat{a}_7\end{bmatrix}_t = \begin{bmatrix}\hat{a}_1\\\hat{a}_2\\\hat{a}_3\end{bmatrix}_{2t} = \hat{a}^{2t}. \qquad (2.3)$$

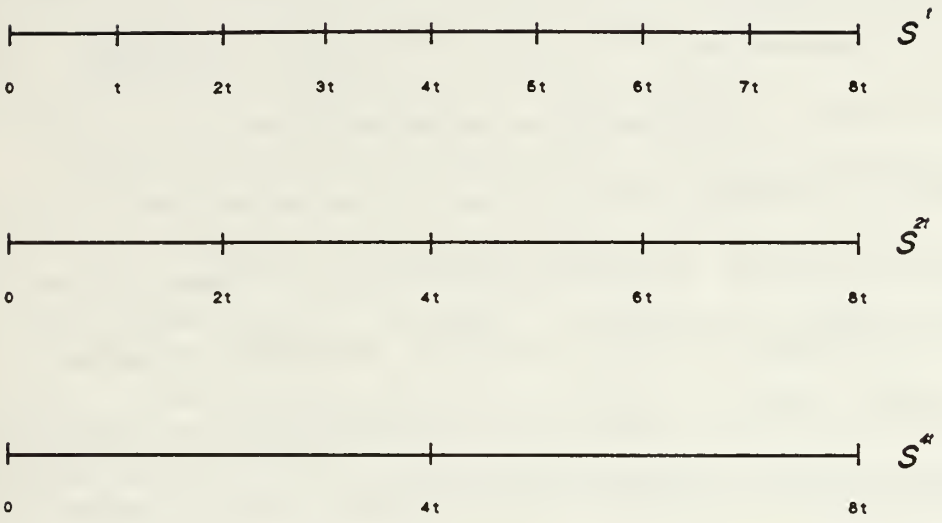

Figure 2.1: Decimation of a Vector on a Fine Grid, S^t, to Coarser Grids, S^{2t} and S^{4t}.

The last intergrid transfer to be discussed deals with transferring matrices between grids or sampling periods. Note that the linear and the full weighting operators are transposes of each other up to a constant. This is an important property when transferring matrices between grids. The matrix intergrid transfer is done using the linear and full weighting operators together such as

$$R^{2t} = I_t^{2t} R^t I_{2t}^t \quad . \tag{2.4}$$

The intergrid transfer operators are essentially interpolation and decimation operators. Reducing the sampling rate or increasing the sampling period of a signal by a factor F is called F:1 decimation. A 2:1 decimation is performed by the full-weighting operator when transferring vectors to coarser grids. Increasing the sampling rate or reducing the sampling period by a factor F is called 1:F interpolation. The linear operator performs a 1:2 interpolation when transferring vectors to finer grids.

The idea of a coarse grid correction may now be discussed [Ref. 7]. After iterating on the fine grid until the convergence slows down or deteriorates, the iteration can be continued by computing the residual as in (2.1). Transferring the residual to a coarser grid then allows further iteration on the low frequency error components. An approximation of the error is obtained on the coarser grid and is transferred back to the original fine grid where the error approximation is used to correct the fine grid solution. Iteration on the fine grid will eliminate the high frequency components of the error, leaving only the low frequency error components.

For the coarse grid correction scheme to work it is important that the error is transferred accurately back to the fine grid. Since the error is smooth or of a low frequency, linear interpolation works very well and the error will be represented accurately on the fine grid [Ref. 7].

1. Toeplitz Approximation Algorithm

One reason for the slow rate of convergence of most of the standard iterative algorithms is that they fail to take advantage of the structure of the autocorrelation matrix R. This matrix is symmetric but not Toeplitz when the covariance method of data formulation is used [Ref. 1]. It does, however, approach a Toeplitz matrix as the number of data points used to form the matrix is increased. Efficient methods such as the Levinson recursion are available to invert a Toeplitz matrix.

An iterative algorithm which takes advantage of the near Toeplitz structure of the autocorrelation matrix R [Refs. 2,5] is developed by splitting R into a Toeplitz matrix T, and a residual matrix S. The matrix T is obtained by averaging the diagonal elements of R. This Toeplitz approximation provides a natural splitting of the autocorrelation matrix, $R = T + S$, which is used to develop an iterative algorithm as follows. Beginning with the system of equations

$$Ra = r \,, \tag{2.5}$$

substituting for $R = T + S$ gives

$$(T + S)a = r \tag{2.6}$$

and, rearranging terms,

$$Ta = r - Sa \,. \tag{2.7}$$

Now, substituting $S = R - T$ in (2.7)

$$Ta = r - (R - T)a \tag{2.8}$$

and solving for a yields

$$a = T^{-1}r - T^{-1}Ra + a \ . \tag{2.9}$$

With this, the iterative algorithm becomes

$$a^{(k+1)} = T^{-1}r - T^{-1}Ra^{(k)} + a^{(k)} \ . \tag{2.10}$$

Letting $a_0 = T^{-1}r$ be the initial estimate of the parameter vector a, then the Toeplitz approximation iterative algorithm has the form

$$a^{(k+1)} = a_0 + a^{(k)} - T^{-1}Ra^{(k)} \ . \tag{2.11}$$

The Toeplitz approximation matrix T may be viewed another way. It can be thought of as a pre-conditioning matrix for R. The difficulty with most iterative methods is that as R becomes increasingly ill-conditioned, the iterative algorithm takes much longer to converge. It has been experimentally observed that the product $T^{-1}R$ does in fact have a better condition number than R by at least an order of magnitude when R is badly conditioned. As a result, the Toeplitz approximation algorithm converges much faster to the true parameter values than most standard iterative algorithms. [Ref. 5]

B. MULTIGRID V CYCLE

The multigrid V cycle (*V-cycle*) can now be developed using the well-defined methods of transferring vectors and matrices from fine to coarse grids and vice-versa, along with the coarse grid correction scheme discussed in the previous section. The best

way to solve the coarse grid correction problem is to use recursion. The coarse grid problem is not much different from the original fine grid problem. Therefore, the coarse grid correction scheme is applied to the residual equation (2.1) on one grid coarser than the fine grid and the process is repeated on successively coarser grids until a direct solution of the residual equation is possible. [Ref. 7]

The *V-cycle* telescopes down to the coarsest grid, which theoretically can be a single point, and then works its way back up to the finest grid (see Figure 2.2). The compact recursive *V-cycle* algorithm is given by:

$$\hat{a}^{l} = MV^{l}(\hat{a}^{l}, r^{l})$$

1. Iterate v times on $R^{l}a^{l} = r^{l}$ with an initial guess $\hat{a}^{l}$.

2. If coarsest grid, then go to step 4.
   Else $r^{2l} = I^{2l}_{l}(r^{l} - R^{l}\hat{a}^{l})$
   $\hat{a}^{2l} = 0$
   $\hat{a}^{2l} = MV^{2l}(\hat{a}^{2l}, r^{2l})$.

3. Correct $\hat{a}^{l} = \hat{a}^{l} + I^{l}_{2l}\hat{a}^{2l}$.

4. Iterate v times on $R^{l}a^{l} = r^{l}$ with initial guess $\hat{a}^{l}$.

The *V-cycle* is just one of a family of multigrid cycling schemes. The entire family is called the μ-*cycle* method where μ would be equal to the number of *V-cycles* to perform. In practice, only $\mu = 1$ (which gives the *V-cycle*) and $\mu = 2$ are used. The resulting method when $\mu = 2$ is called the *W-cycle*. [Ref. 7]

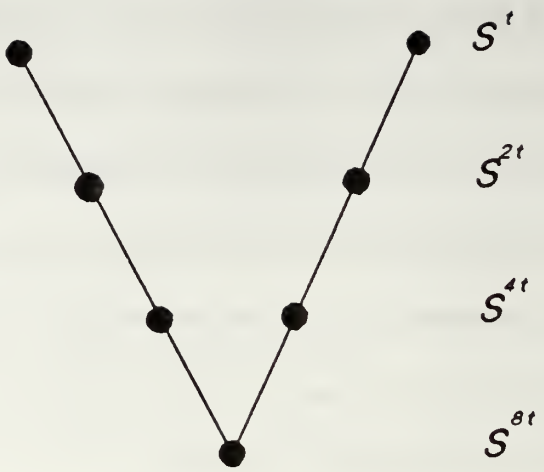

Figure 2.2: Schedule of Grids for the *V-cycle* on Four Levels.

C. FULL MULTIGRID V CYCLE

The coarse grid correction scheme developed in section B led directly to the *V-cycle*. Yet another scheme to be explored is based on the nested iteration idea. A method used to improve the convergence of an iterative method is to begin the iteration with a better guess of the correct solution. A good strategy of obtaining a better initial guess is to first iterate the solution on a coarser grid. This provides a better guess for the initial parameter vector and is computationally inexpensive compared to the fine grid problem. Therefore, nested iteration uses coarse grids to obtain improved initial guesses for fine grid problems.

To obtain an improved initial guess for the *V-cycle*, one would solve the problem on a coarse grid one step below the fine grid. The initial guess for a coarser grid is obtained by solving the problem on the grid one step coarser than the current grid.

Thus, the nested iteration leads ultimately to the coarsest grid as a starting point which leads directly to the full multigrid V cycle (*FMV-cycle*). Using and expanding on the ideas presented in the *V-cycle* and incorporating the idea of nested iteration, the *FMV-cycle* algorithm is given by:

$$\hat{a}^t = FMV^t(\hat{a}^t, r^t)$$

1. If coarsest grid, then go to step 3.
   Else $r^{2t} = I^{2t}_t(r^t - R^t\hat{a}^t)$
   $\hat{a}^{2t} = 0$
   $\hat{a}^{2t} = FMV^{2t}(\hat{a}^{2t}, r^{2t})$.

2. Correct $\hat{a}^t = \hat{a}^t + I^t_{2t}\hat{a}^{2t}$.

3. $\hat{a}^t = MV^t(\hat{a}^t, r^t)$ υ times.

The scheduling of grids in the *FMV-cycle* begins at the coarsest grid where the iteration begins (See Figure 2.3). A single *V-cycle* is performed at the next finer grid.

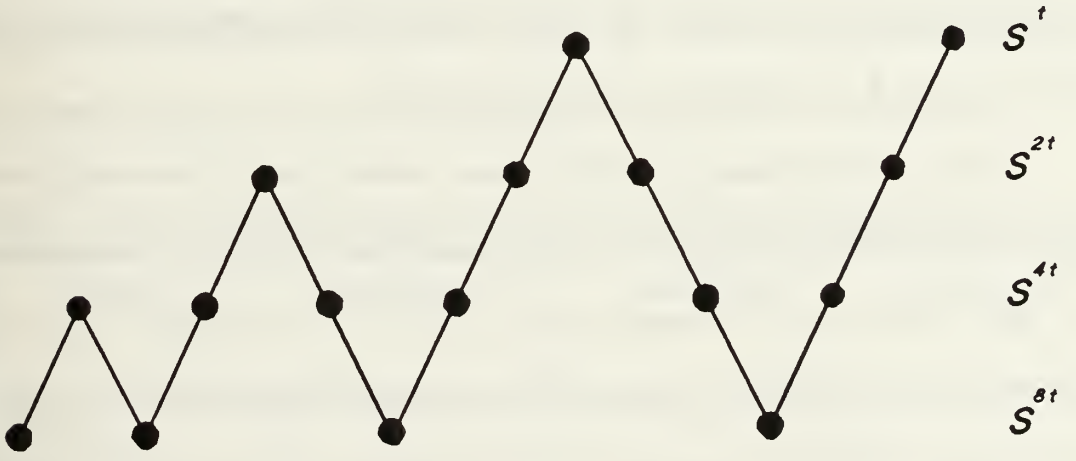

Figure 2.3: Schedule of Grids for the *FMV-cycle* on Four Levels.

This continues as each *V-cycle* is preceded by a smaller one designed to provide the best initial guess possible to the next finer grid. The extra work done in these preliminary *V-cycles* is not overly expensive and greatly enhances the performance since the low frequency components of the error are dealt with directly on the coarse grids. The *FMV-cycle* is the culmination of ideas and techniques which individually have been known and used for a long time. The result is a very simple but powerful scheme that can be used to improve the convergence properties of most iterative algorithms. [Ref. 7]

D. PERFORMANCE COMPARISONS

The preceding section was devoted to the development of the multigrid cycling schemes. The practical issue of complexity and performance will now be addressed. For convenience, the computational cost of multigrid methods is estimated in terms of *work units* (WU). A work unit, WU, is the cost of performing one iteration on the finest grid. The number of fine grid iterations performed will be referred to using the variable k. It is customary to neglect the cost of the intergrid transfer operations which could amount to as much as 15-20 percent of the entire cycle.

Consider a d-dimensional grid with N^d points, where $N = 2^n$. An example would be a one-dimensional modeling problem with a parameter vector of $N=16$ points, i.e. $d=1$, $n=4$. With one iteration on each grid level, i.e., $\nu = 1$, each grid level in a single *V-cycle* is visited twice (once at each grid level as the algorithm telescopes its way down to the coarsest grid and once at each grid level on the way back up to the finest grid).

Each grid S^{p^l} requires p^{-d} of a work unit. Adding these costs from each grid level gives the *V-cycle* computation cost as

$$2(1+2^{-d}+2^{-2d}+\ldots+2^{-nd}) < \frac{2}{1-2^{-d}} \quad WU. \qquad (2.12)$$

Therefore a single *V-cycle* costs 4 WU for a one-dimensional problem; the cost is 8/3 WU for $d = 2$ and 16/7 WU for $d = 3$. [Ref. 7]

The computational cost calculation for the *FMV-cycle* is similar. Again, using only one iteration on each grid level, i.e. $v = 1$, an *FMV-cycle* beginning from the S^1 grid costs $2(1 - 2^{-d})^{-1}$ WU. A *V-cycle* beginning from S^{2^l} costs 2^{-d} of a full *V-cycle* and in general, a *V-cycle* beginning from S^{p^l} costs p^{-d} of a full *V-cycle*. Summing these costs gives the *FMV-cycle* computation cost

$$\left(\frac{2}{1-2^{-d}}\right)(1+2^{-d}+2^{-2d}+\ldots+2^{-nd}) < \frac{2}{(1-2^{-d})^2} \quad WU. \qquad (2.13)$$

For the one-dimensional problem, a single *FMV-cycle* costs 8 WU, the cost is about 7/2 WU for $d = 2$ and 5/2 WU for $d = 3$ [Ref. 7]. Table 2.1 displays the previously mentioned computational costs of the multigrid cycles in terms of work units.

TABLE 2.1: COMPUTATIONAL COST OF A SINGLE MULTIGRID CYCLE

Method ($v = 1$)	Work Units ($d = 1$)	Work Units ($d = 2$)	Work Units ($d = 3$)
Iterative	1	1	1
V-cycle	4	8/3	16/7
FMV-cycle	8	7/2	5/2

The more interesting and more important comparison between the straight iterative method and the multigrid cycles is the effectiveness of the technique versus its computational complexity. As a typical result, take a one-dimensional case with $N^d=16$ grid points and $n=4$. In order to achieve similar results, the straight iterative method required $k=12$ fine grid iterations. Using $\upsilon=3$, i.e., 3 iterations were done at each grid level, the *V-cycle* technique required $k=6$ fine grid iterations and the *FMV-cycle* technique required less than 1 fine grid iteration. This means the *FMV-cycle* technique had converged to a similar solution before having conducted its first fine grid iteration. The computational complexity comparisons of this example are shown in Table 2.2.

TABLE 2.2: EFFECTIVENESS VS. COMPUTATIONAL COMPLEXITY

Method (υ = 3)	Fine Grid Iterations, k	Work Units (WU's)
Iterative	12	12
V-cycle	6	11.63
FMV-cycle	< 1	10.89

Although Table 2.1 shows that a single multigrid cycle is computationally more costly than its iterative counterpart, the results displayed in Table 2.2 provide a very different conclusion. The multigrid cycles are more efficient per iteration [Ref. 7] and thus can be computationally less expensive in certain instances. It should be noted that the savings will vary according to the specifics of each problem. As a general rule, the greater the number of points in the grid, the greater the effectiveness of the multigrid techniques and thus the greater the savings will be. Also, the *FMV-cycle* is usually

preferable over a single *V-cycle*. Chapter IV deals more specifically with the issue of effectiveness using the system modeling algorithms discussed in Chapter III with varying grid sizes.

III. SYSTEM MODELING

A. THE FINITE IMPULSE RESPONSE (FIR) MODEL

We are trying to model a system with input $x(n)$ where $y(n)$ is the output of the system and $\hat{y}(n)$ is the output of the M^{th}-order model (see Figure 3.1). The output at time

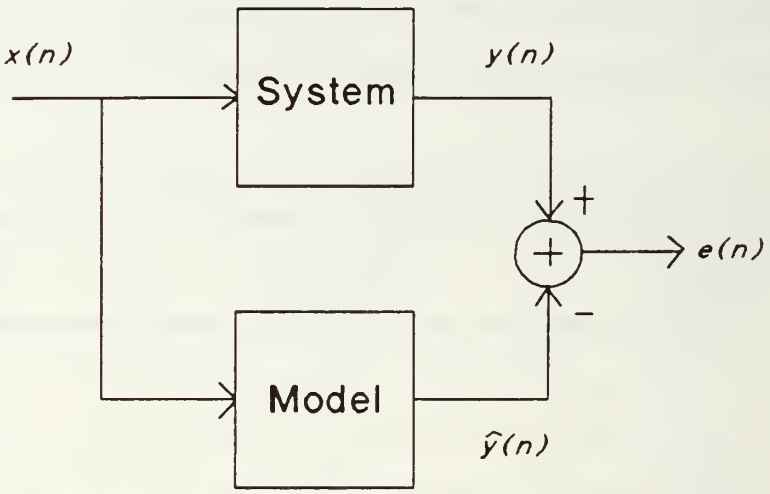

Figure 3.1: Input and Output Relationships of System Modeling.

n is simply a weighted linear combination of the input:

$$\hat{y}(n) = x(n)b_0^M + x(n-1)b_1^M + \cdots + x(n-M)b_M^M . \qquad (3.1)$$

This can be written as

$$\hat{y}(n) = \boldsymbol{x}_n \boldsymbol{b}^M , \qquad (3.2)$$

where $x_n = [x(n)\ x(n\text{-}1) \cdots x(n\text{-}M)]$ and $b^M = [b_0^M\ b_1^M \cdots b_M^M]^T$ are the (1 x M+1) input data vector and the (M+1 x 1) vector containing the filter weights, or coefficients, respectively. The filter output at time n is $\hat{y}(n)$. The superscript M indicates that these are the coefficients of an M^{th}-order filter. Because the true value of b^M is not known, equation (3.2) produces an estimate for the output, $\hat{y}(n)$. Consequently, the error $e(n)$ between the system output $y(n)$ and the FIR filter output $\hat{y}(n)$, is given by

$$e(n) = y(n) - \hat{y}(n) = y(n) - \boldsymbol{x}_n \boldsymbol{b}^M . \qquad (3.3)$$

We seek a least squares solution to (3.2) that minimizes the sum of squared errors. The key is to form an overdetermined set of P+1 equations (where $P \geq M$ is necessary for a unique solution) [Ref. 5] which can be written compactly in matrix notation as

$$\hat{y} = \boldsymbol{X}\boldsymbol{b}^M \qquad (3.4)$$

where $X = [x_n^T\ x_{n\text{-}1}^T \cdots x_{n\text{-}P}^T]^T$.

From (3.4), the error vector can be written as: $e = y - \hat{y} = y - Xb^M$. The sum of squared errors ζ is given by

$$\zeta = \sum_{j=n-P}^{n} |\boldsymbol{e}(j)|^2 = \boldsymbol{e}^T\boldsymbol{e} = \boldsymbol{y}^T\boldsymbol{y} - \boldsymbol{y}^T\boldsymbol{X}\boldsymbol{b}^M - \boldsymbol{b}^{M^T}\boldsymbol{X}^T\boldsymbol{y} + \boldsymbol{b}^{M^T}\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{b} . \qquad (3.5)$$

The minimization of ζ is accomplished by taking its derivative with respect to the filter coefficient vector b^M [Ref. 1,5,10,11] and setting the resultant to zero:

$$\frac{\partial(\zeta)}{\partial(\boldsymbol{b}^M)} = 0 - 2\boldsymbol{X}^T\boldsymbol{y} + 2\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{b}^M = 0 \ . \qquad (3.6)$$

Rearranging the terms in (3.6) gives

$$\boldsymbol{X}^T\boldsymbol{X}\boldsymbol{b}^M = \boldsymbol{X}^T\boldsymbol{y} \qquad (3.7)$$

as the requisite condition for the sum of squared errors to be minimum. Solving (3.7) yields the optimal FIR filter coefficients

$$\boldsymbol{b}^M = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \ . \qquad (3.8)$$

A more standard representation of (3.7), referred to as the *normal equations* is

$$\boldsymbol{R}\boldsymbol{b}^M = \boldsymbol{r} \ , \qquad (3.9)$$

where $\boldsymbol{R} = \boldsymbol{X}^T\boldsymbol{X}$ is the correlation matrix, and $\boldsymbol{r} = \boldsymbol{X}^T\boldsymbol{y}$ is the cross-correlation of the input data matrix $\boldsymbol{X}$ and the output data vector y. Theoretically, if $\boldsymbol{R}$ is of full rank, the solution to (3.9) is given by $\boldsymbol{b}^M = \boldsymbol{R}^{-1}\boldsymbol{r}$. This thesis develops iterative algorithms for FIR, AR, IIR and 2-D AR models utilizing multigrid techniques which are alternatives to the direct inversion of $\boldsymbol{R}$.

B. THE AUTOREGRESSIVE (AR) MODEL

AR modeling is approached through the related process of linear prediction [Ref. 1]. The output of an *Nth* order linear predictor is given by

$$\hat{x}(n) = -a_1 x(n-1) - a_2 x(n-2) - \cdots - a_N x(n-N) \qquad (3.10)$$

where a_i are the linear predictor coefficients, N is the linear predictor order and $x(n-i)$ for $i = 1, 2, \cdots, N$ are the inputs to the linear predictor. The error can then be written as

$$e(n) = x(n) - \hat{x}(n) = \sum_{k=0}^{N} a_k x(n-k) \ , \qquad (3.11)$$

where $a_0 = 1$. To formulate the least squares version of the problem, define the vector

$$\boldsymbol{x}_0 = \begin{bmatrix} x(n_I) \\ x(n_I+1) \\ \vdots \\ \vdots \\ x(n_F) \end{bmatrix} \qquad (3.12)$$

where n_I and n_F are some initial and final values of n that define the interval over which the minimization is performed. The vector of estimates can be expressed as

$$\hat{\boldsymbol{x}}_0 = -\boldsymbol{X}'\boldsymbol{a}' \qquad (3.13)$$

where

$$\boldsymbol{X}' = \begin{bmatrix} x(n_I-1) & x(n_I-2) & \cdots & x(n_I-N) \\ x(n_I) & x(n_I-1) & \cdots & x(n_I-N+1) \\ \vdots & \vdots & & \vdots \\ x(n_F-1) & x(n_F-2) & \cdots & x(n_F-N) \end{bmatrix} \qquad (3.14)$$

and

$$\mathbf{a}' = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix} . \tag{3.15}$$

This problem is a special case of the Wiener filtering problem [Ref. 1], therefore the solution here can be written as

$$\mathbf{a}' = -(\mathbf{X}'^T\mathbf{X}')^{-1}\mathbf{X}'^T\mathbf{x}_0 \tag{3.16}$$

where $(X'^TX')^{-1}X'^T$ is the pseudo-inverse of the data matrix X' [Ref. 1].

In order to produce a set of *normal equations*, the error vector is defined as

$$\epsilon = \mathbf{x}_0 - \hat{\mathbf{x}}_0 . \tag{3.17}$$

It then follows that

$$[\, \mathbf{x}_0 \;\vdots\; \mathbf{X}' \,] \begin{bmatrix} 1 \\ \cdots \\ \mathbf{a}' \end{bmatrix} = \mathbf{X}\mathbf{a} = \epsilon \tag{3.18}$$

where

$$\mathbf{a} = \begin{bmatrix} 1 \\ \mathbf{a}' \end{bmatrix} = \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} . \tag{3.19}$$

From the orthogonality theorem [Ref. 1] we can write

$$\boldsymbol{X}^T\boldsymbol{e} = \begin{bmatrix} \boldsymbol{x}_0^T \\ \boldsymbol{X}'^T \end{bmatrix} \boldsymbol{e} = \begin{bmatrix} S \\ 0 \end{bmatrix} \tag{3.20}$$

where S is the sum of squared errors. Combining equations (3.18) and (3.20) produces the normal equations

$$\boldsymbol{R}\boldsymbol{a} = \begin{bmatrix} S \\ 0 \end{bmatrix} \tag{3.21}$$

where $\boldsymbol{R} = \boldsymbol{X}^T\boldsymbol{X}$ is the correlation matrix.

There are two well-known methods for linear prediction which result in two different choices for n_I and n_F (see (3.12)). They are the autocorrelation method and the covariance method. The discussion of these methods will be limited to the covariance method here.

In the covariance method the values of $n_I = N$ and $n_F = L$ are chosen where L is the number of samples in the data sequence to be modeled by the *N-th* order prediction error filter. The data matrix has the form

$$\boldsymbol{X} = \begin{bmatrix} x(N) & x(N-1) & \cdots & x(0) \\ \vdots & \vdots & & \vdots \\ x(L-1) & x(L-2) & \cdots & x(L-N-1) \end{bmatrix}. \tag{3.22}$$

The resulting correlation matrix is symmetric and positive semidefinite but it is not Toeplitz. The covariance method is often preferred because it makes use of only the measured data, thus eliminating the windowing effects and frequently results in a more accurate model. [Ref. 1]

C. THE INFINITE IMPULSE RESPONSE (IIR) MODEL

The infinite impulse response (IIR) model is more complicated than the two previously presented models. As in the FIR case, we are trying to model a system whose input is $x(n)$. The output of the system is $y(n)$ and the output of model is $\hat{y}(n)$ (See Figure 3.1). The filter output $\hat{y}(n)$ is a linear combination of the input sequence $x(n)$, $x(n\text{-}1)$, $\cdots$, $x(n\text{-}M)$, as well as the previous output samples $\hat{y}(n\text{-}1)$, $\hat{y}(n\text{-}2)$, $\cdots$, $\hat{y}(n\text{-}N)$, where M and N are the orders of the input and output coefficients, respectively. As in the FIR case, start with the difference equation representation:

$$\begin{aligned}\hat{y}(n) &= \hat{y}(n-1)\,a_1^N + \hat{y}(n-2)\,a_2^N + \cdots + \hat{y}(n-N)\,a_N^N \\ &\quad + x(n)\,b_0^M + x(n-1)\,b_1^M + \cdots + x(n-M)\,b_M^M ,\end{aligned} \tag{3.23}$$

which can be written as

$$\begin{aligned}\hat{y}(n) &= \left[\, \hat{\boldsymbol{y}}_{n-1} \;\vdots\; \boldsymbol{x}_n \,\right] \begin{bmatrix} \boldsymbol{a}^N \\ \cdots \\ \boldsymbol{b}^M \end{bmatrix} \\ \hat{y}(n) &= \boldsymbol{z}_n \theta ,\end{aligned} \tag{3.24}$$

where z_n is a 1 x $N+M+1$ data vector and θ is a $N+M+1$ x 1 vector containing the IIR filter weights. Because the true values of a^N and b^M are not known, equation (3.24) will only produce an estimate for the output, $\hat{y}(n)$. Thus, the error between the true output $y(n)$ and the IIR filter output $\hat{y}(n)$ is given by

$$e(n) = y(n) - \hat{y}(n) = y(n) - \boldsymbol{z}_n\theta \ . \qquad (3.25)$$

As in section A, the least squares solution to (3.24) for θ comes from an over-determined set of $P+1$ equations (where $P \geq N+M$ is necessary for a unique solution) [Ref. 5]. In matrix notation it is given by

$$\hat{y} = \boldsymbol{Z}\theta \ , \qquad (3.26)$$

where the data matrix $\boldsymbol{Z} = [z_n{}^T z_{n-1}{}^T \cdots z_{n-P}{}^T]^T$. The error vector is defined as $\boldsymbol{e} = \boldsymbol{y} - \boldsymbol{Z}\theta$. Minimizing the sum of the squared errors, similar to equations (3.5 - 3.7), gives

$$\boldsymbol{Z}^T\boldsymbol{Z}\theta = \boldsymbol{Z}^T\boldsymbol{y} \ . \qquad (3.27)$$

The *normal equation* representation of (3.27) can be written as

$$\boldsymbol{R}\theta = \boldsymbol{r} \ , \qquad (3.28)$$

where $\boldsymbol{R} = \boldsymbol{Z}^T\boldsymbol{Z}$ is referred to as the correlation matrix, and $\boldsymbol{r} = \boldsymbol{Z}^T\boldsymbol{y}$ is the cross-correlation vector.

D. THE 2-D AUTOREGRESSIVE (AR) MODEL

The 2-D AR model assumes a stationary random process $x(n_1,n_2)$ that is the output of an AR filter excited by a white noise input $w(n_1,n_2)$ having a variance $\sigma_w{}^2$. The AR model is an all pole filter with a region of support A over which the parameters $a(k_1,k_2)$ are non-zero. The difference equation for the system that generates $x(n_1,n_2)$ can be expressed as

$$x(n_1,n_2) = -\sum_{(i,j)}\sum_{\in A} a(i,j)x(n_1-i,n_2-j)+w(n_1,n_2) \quad . \qquad (3.29)$$

In this thesis we only discuss the *quarter-plane* support and the *combined quadrant* support which were investigated in this research. Multiplying both sides of (3.29) by $x(n_1-l_1,n_2-l_2)$ and computing the statistical expectation of the resulting expression [Refs. 6,8] leads to the *normal equation*

$$R_x(l_1,l_2) = -\sum_{(i,j)}\sum_{\in A} a(i,j)R_x(l_1-i,l_2-j) \quad , \qquad (3.30)$$

which holds for $l_1,l_2 \geq 0$. The coefficients $a(i,j)$ can be derived from this normal equation. The structure of the normal equation depends on the region of support A.

1. Quarter-Plane Support

The region A is said to have quarter-plane (QP) support when $a(i,j)$ are non-zero in one quadrant only. For QP support the normal equation has the form [Ref. 8]

$$R_x(l_1,l_2) = \sum_{i=0}^{P_1-1}\sum_{j=0}^{P_2-1} a(i,j)R_x(l_1-i,l_2-j), \quad (i,j)\neq(0,0) \qquad (3.31)$$

where $l_1 = 1,2,\ldots,P_1\text{-}1$ and $l_2 = 1,2,\ldots,P_2\text{-}1$ with P_1 and P_2 being the dimensions of A. If we assume that $a(0,0) = 1$, then (3.31) may be expressed in a *block matrix* form as

$$\begin{bmatrix} R_0 & R_{-1} & R_{-2} & \cdots & R_{-P_1+1} \\ R_1 & R_0 & R_{-1} & \cdots & R_{-P_1+2} \\ R_2 & R_1 & R_0 & \cdots & R_{-P_1+3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R_{P_1-1} & R_{P_1-2} & R_{P_1-3} & \cdots & R_0 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \vdots \\ a_{P_1-1} \end{bmatrix} = \begin{bmatrix} \epsilon^{(0)} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} . \qquad (3.32)$$

Iterative algorithms utilizing the multigrid technique are presented in Chapter IV to solve these equations more efficiently. Details of each block in the matrix and the error variance vector $\epsilon^{(0)}$ can be found in [Refs. 6,8].

2. 2-D AR Spectral Estimation

The 2-D AR power spectral estimate is given by

$$\hat{P}_x(\omega_1, \omega_2) = |H(\omega_1, \omega_2)|^2 P_\omega , \qquad (3.33)$$

where P_ω is the power spectral density of the input and $H(\omega_1,\omega_2)$ is the transfer function of the 2-D AR model [Refs. 6,8]. If the input is white noise with a constant power spectrum of amplitude σ_ω^2, then (3.33) can be written as

$$\hat{P}_x(\omega_1, \omega_2) = \frac{\sigma_\omega^2}{|1 + \sum\sum_{(k_1, k_2 \in A)} a(k_1, k_2)\, e^{-j(\omega_1 k_1 + \omega_2 k_2)}|^2} . \qquad (3.34)$$

The 2-D AR spectral estimate is used to study the performance of the 2-D AR algorithm.

E. INTEGRATING MULTIGRID WITH THE MODELING ALGORITHMS

The previous sections develop the theory of the modeling algorithms used to test the performance of the multigrid techniques in a digital signal processing (DSP) application. In Chapter II, the multigrid theory and the Toeplitz approximation iterative

algorithm is presented. We now turn to the application of the multigrid techniques to the algorithms that have been discussed.

The first step is to develop multigrid programs. These are developed from simple iterative algorithms that they are designed to enhance in a modular fashion. Modular programs are easier to verify and debug. The various components of the program, (i.e. the modeling algorithm, the iterative algorithm, the *V-cycle* algorithm, the *FMV-cycle* algorithm, the residual computation, and the interpolation and restriction operator functions), can be mixed and matched and replaced individually.

These multigrid techniques can be applied with varying degrees of success to any problem that can be formulated in the form $\boldsymbol{Ra} = \boldsymbol{r}$ discussed in Chapter II. Each application works best if it has its own calling routine that can be tailored specifically to the problem at hand. For example, calling routines in this thesis have been written to perform system modeling on FIR, AR, IIR, and 2-D AR systems. Each of these modeling problems can be solved using the straight iterative algorithm, the V-cycle method, or the *FMV-cycle* method.

Sufficient error checking must be done to ensure that the filter order is compatible with the intergrid transfer functions, e.g., the parameter vector length must be odd. This in itself can be a drawback of multigrid and is further discussed in Chapter V. Calls to the modeling algorithm then take the data sequence and compute the correlation matrix, $\boldsymbol{R}$, and the cross-correlation vector, $\boldsymbol{r}$. Once this preliminary processing is complete, the appropriate multigrid subroutines are called to solve for the parameter vector, $\boldsymbol{a}$.

The core of the multigrid technique in any application consists of the *V-cycle* and the *FMV-cycle* recursive algorithms. It is important to note that the recursive nature of these definitions absolutely requires that they be written as subroutines and thus leads to the modular nature of the multigrid programs. These routines can be implemented exactly as stated in the recursive definitions presented in Chapter II. The key is to substitute the iterative algorithm that works best in your application wherever the definition requires that an iteration or relaxation takes place.

The *V-cycle* and *FMV-cycle* algorithms do not stand alone, however. Both must call supporting subroutines. Each of the multigrid cycles require that the residual, d, be computed somewhere in the routine. Once the residual is computed, the correlation matrix, R, the cross-correlation vector, r, and the residual, d, must be transferred to a coarser grid; this requires use of the intergrid transfer subroutines. The intergrid transfer functions may also be implemented directly as discussed in Chapter II.

Each application of the multigrid technique has its own associated peculiarities. When applying the multigrid technique to any problem, it is helpful to keep in mind the overall goal which is to solve for the coefficients or parameters, a, in a system of equations of the form $Ra = r$. The theoretical basis behind each modeling problem determines how the correlation matrix, R, and the cross-correlation vector, r, are to be computed. Once that is done, each problem looks very much alike except for the application specific iterative algorithm used.

IV. RESULTS OF SIMULATIONS

A. FIR SIMULATIONS

In order to conduct the following computer simulations, a suitable reference system is required. Given a generated input and output data sequence, the algorithms described in Chapter III could then be used to model or "identify" the system. For the FIR simulations the output data sequence, y, was generated by filtering a zero-mean, unit variance, white noise input sequence, x, with a 22^{nd} order elliptical IIR filter. The selected IIR filter was of a bandpass design with cutoff frequencies of 0.3π and 0.7π. The bandpass design was chosen in order to provide a sufficiently long impulse response. The IIR filter impulse and frequency response are shown in Figure 4.1.

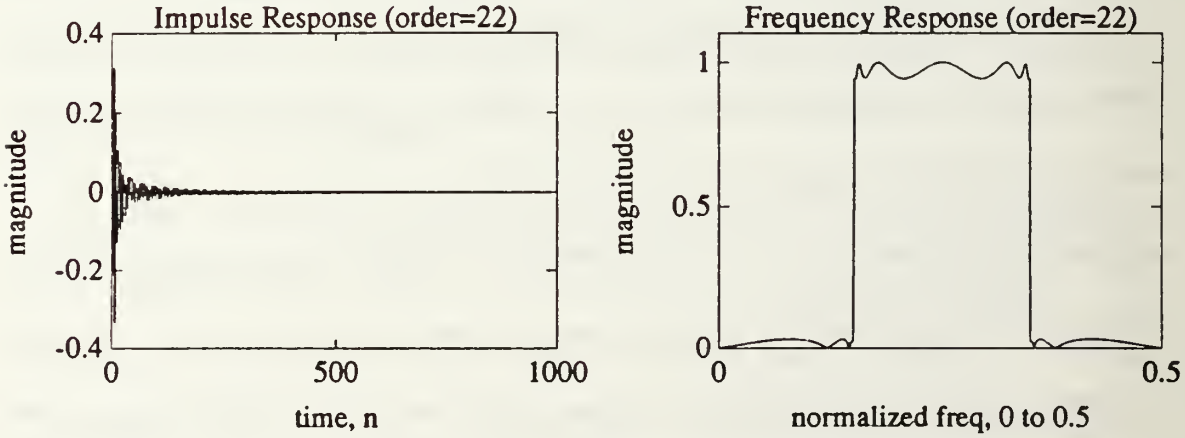

Figure 4.1: 22^{nd} Order Digital Elliptical IIR Filter Impulse and Frequency Response (cutoff frequencies 0.3π & 0.7π)

FIR filters of orders ranging from 31 to 255 were simulated. Only the results of the 127^{th} order ($M=127$) simulations are presented here. Figures A.1 - A.12 in

Appendix A present the remaining FIR simulations. To compare the relative performance, results of the straight iterative algorithm, the *V-cycle*, and the *FMV-cycle* methods are presented for the 127^{th} order model. The relative effectiveness of each method is shown by running the same number of fine grid iterations and comparing the results of each method. In each FIR simulation the number of data points, L, used is equal to 1000. This method of comparison is used for this and all remaining simulations presented in this chapter.

Figure 4.2 shows the results of the straight iterative method. Note that the straight iterative method converges in about four fine grid iterations. It converges rather quickly with a noticeable change in the parameter values for the first few iterations. Notice the normalized frequency response at the first iteration is dominated by a spurious peak at a very low frequency. This is because of the less than optimal parameter values at the first iteration. By the sixth iteration all parameter values are steady and the impulse and frequency response results plots are fairly close to their true values.

The *V-cycle* simulation is presented in Figure 4.3 using the same format to display the results. Note that the change in the parameter values in the first few fine grid iterations is noticeably smaller. This method reaches the solution one fine grid iteration before the straight iterative method and ultimately converges to the same impulse response at the sixth iteration. Of greater interest is the noticeable improvement of the normalized frequency response at the first fine grid iteration.

The *FMV-cycle* simulation is shown in Figure 4.4. The change in parameter values is almost non-existent and the coefficients have certainly converged by the second fine

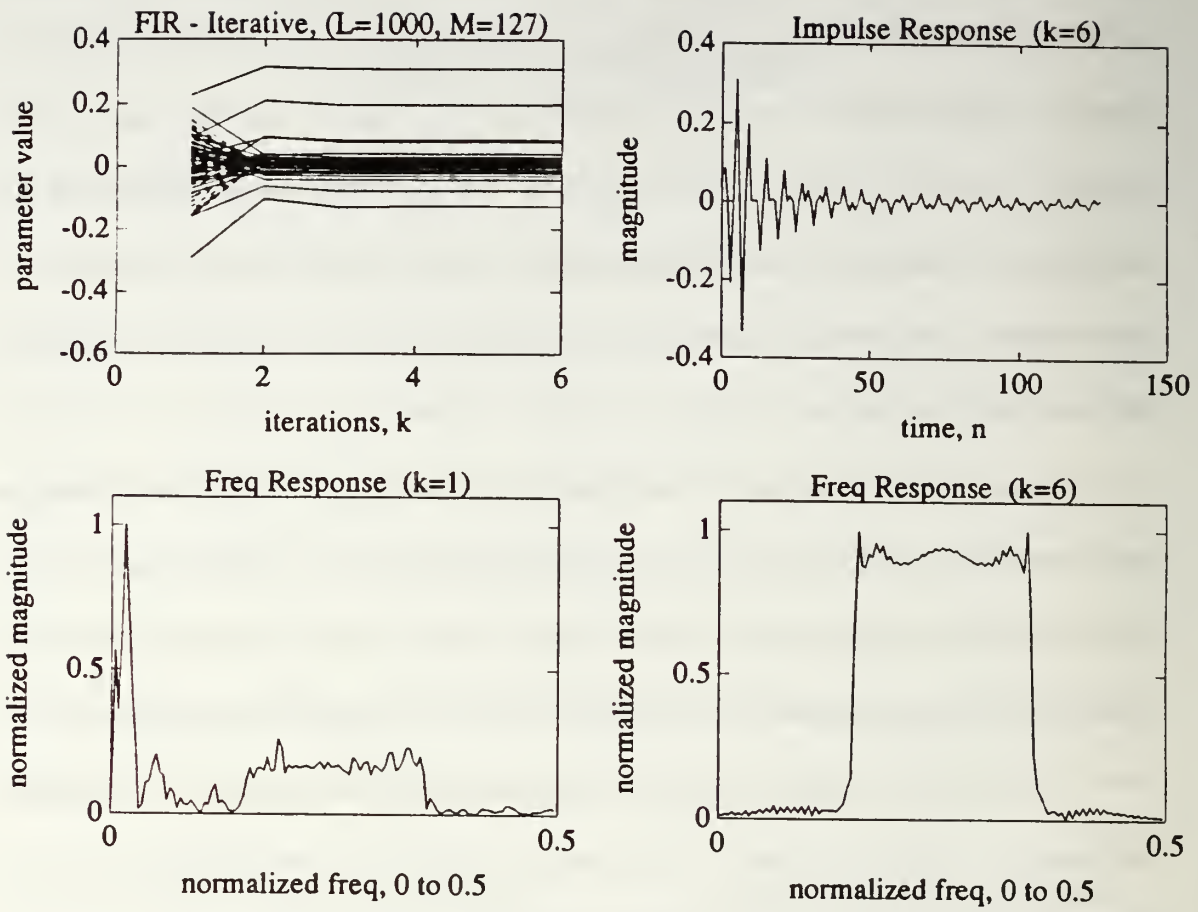

Figure 4.2: 127th Order FIR Model Using Straight Iterative Method. Impulse and normalized frequency responses are shown using $L=1000$ data points.

grid iteration. The normalized frequency response at the first fine grid iteration is much superior to the first two methods presented. The corresponding impulse and frequency responses at the sixth iteration are equivalent to the straight iterative and the *V-cycle* methods. The results illustrate the fast convergence performance of the *FMV-cycle*.

To show that all three methods converge to nearly the same solution, a comparison of impulse response errors is displayed in Figure 4.5. The purpose of this comparison is to show that the multigrid techniques do not sacrifice accuracy for computational

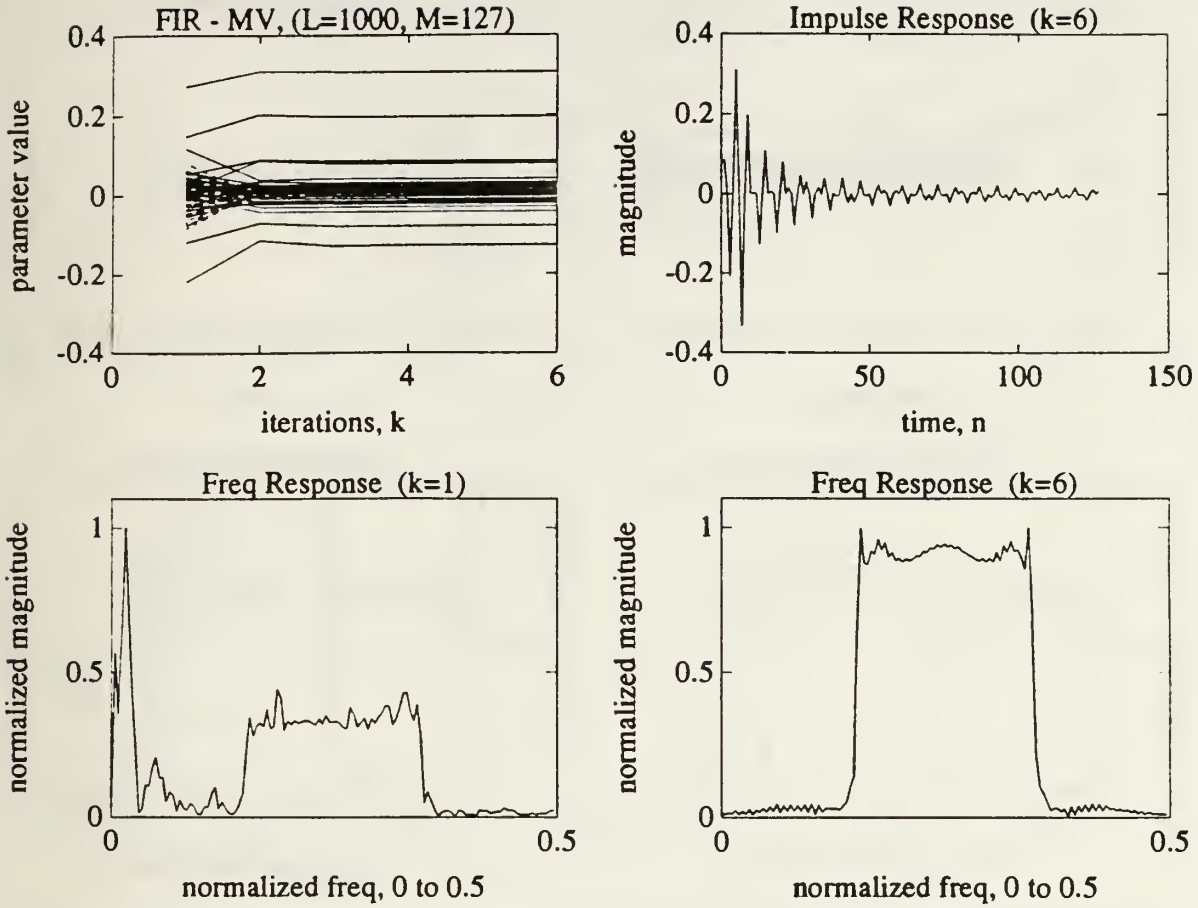

Figure 4.3: 127th Order FIR Model Using *V-cycle* Method. Impulse and normalized frequency responses are shown using $L=1000$ data points.

complexity and indeed enhance the speed of convergence to the correct solution. For the 127th order FIR model case, the impulse responses of the three methods at the sixth fine grid iteration are very close to the true impulse response of the 22nd order IIR elliptical

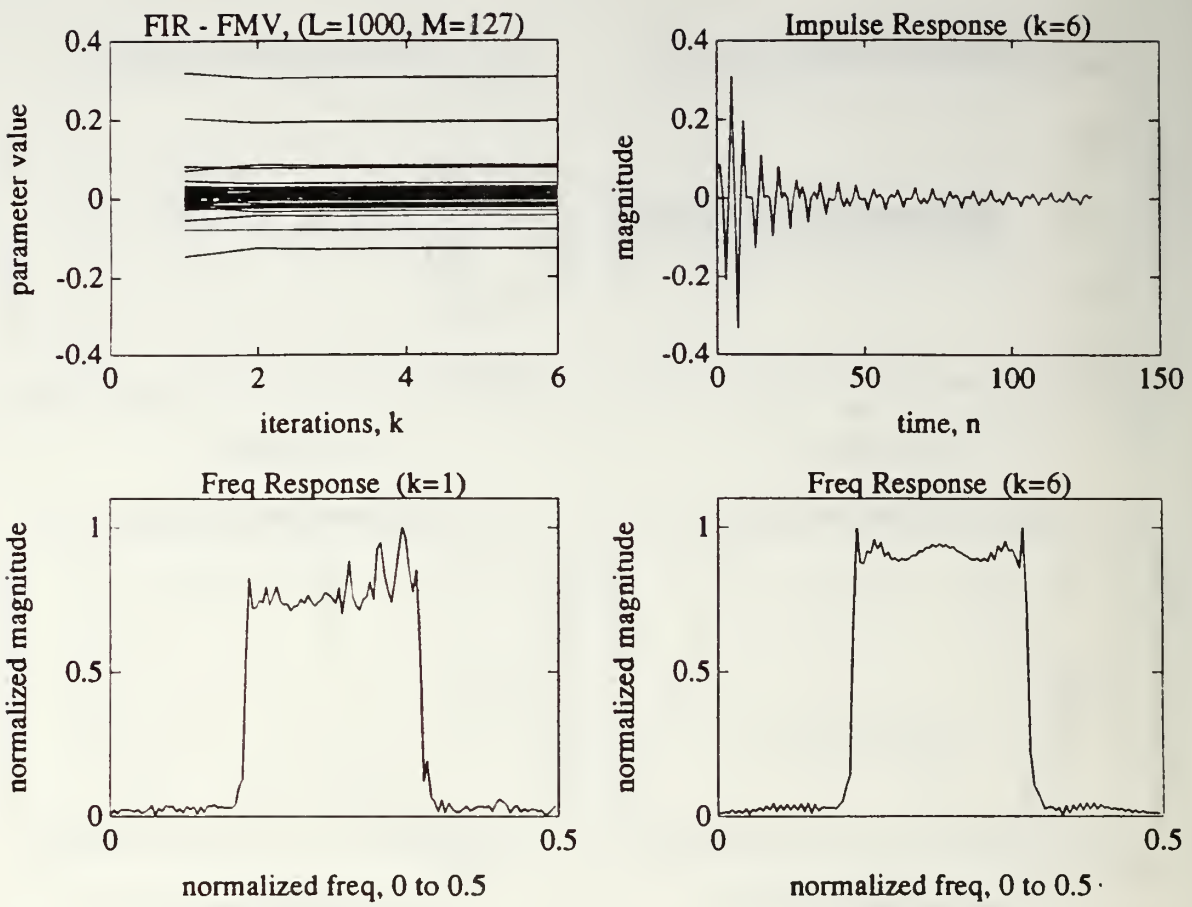

Figure 4.4: 127th Order FIR Model Using *FMV-Cycle* Method. Impulse and normalized frequency response are shown $L=1000$ data points.

bandpass filter. The impulse response error of the two multigrid methods can be seen to be smaller than the straight iterative method because of the enhanced convergence rate.

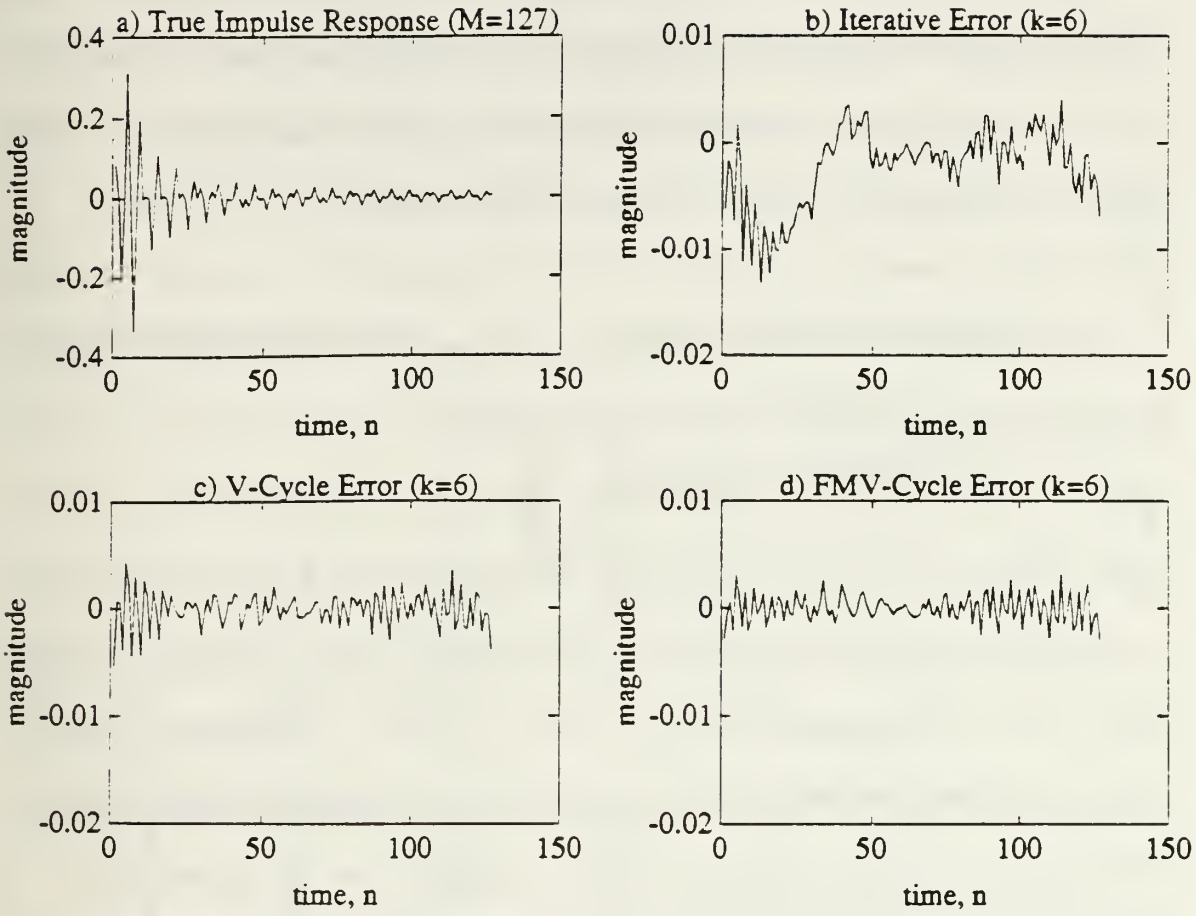

Figure 4.5: 127th Order FIR Model Impulse Response Error Comparison. a) True System Impulse Response; b) Iterative Method Error; c) *V-cycle* Method Error; d) *FMV-cycle* Method Error.

B. AR SIMULATIONS

For the AR model simulations, an $L=300$ sample frame of voiced speech was selected to process for comparison purposes. The AR covariance method modeling algorithm developed in Chapter III was used in these tests. This method is also known as Linear Predictive Coding (LPC) in speech processing literature [Ref. 9]. The speech was sampled at 10 kHz resulting in a frame length of 30 ms.

As in the FIR case, only one representative case is presented here (the 15th order or $N=15$ model). Simulations ranging from 3rd to 63rd order models were conducted and the results of these additional cases are included in Appendix B (see Figures B.1 - B.12). Figure 4.6 displays the results of the straight iterative method.

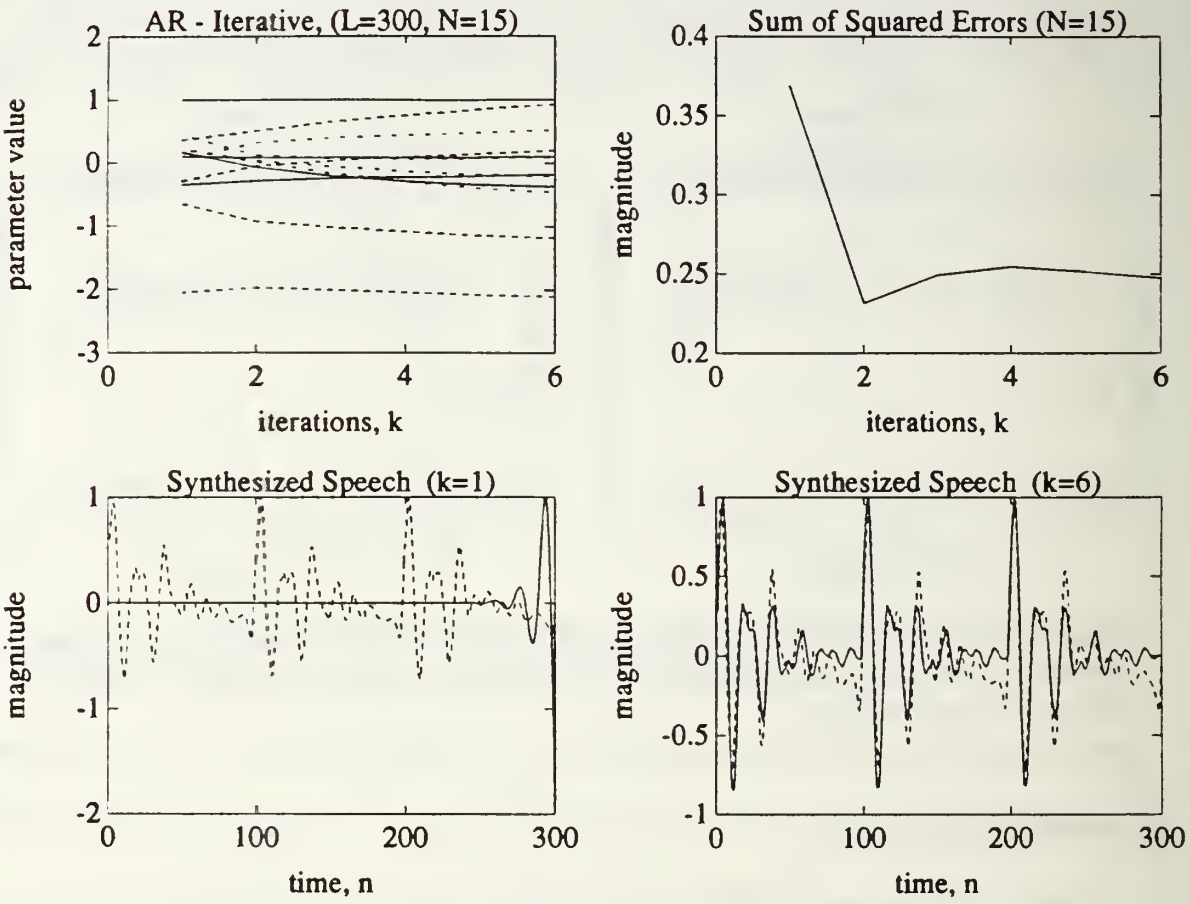

Figure 4.6: 15th Order AR Model of Voiced Speech Using Straight Iterative Method. Original speech waveform shown with dashed lines, synthesized speech waveform shown with solid lines for $L=300$ data points.

The LPC parameters converge very quickly because of the low order model. The sum of squared errors, S, decreases rapidly from 0.37 to approximately 0.25 at the sixth fine grid iteration. The original speech waveform is shown with dashed lines.

Synthesized speech waveforms are shown in solid lines for two cases after completing one and six iterations. It can be seen that no intelligible speech is synthesized at the first iteration as compared to the fairly good speech synthesized at the sixth iteration.

Using the same 300 sample speech signal, the *V-cycle* method produces the results shown in Figure 4.7. The results of the *V-cycle* method are almost identical to the straight iterative method. Three fine grid iterations ($\upsilon=3$) are conducted prior to transferring to a coarser grid for the telescoping *V-cycle* and the coarse grid correction. Because of these three fine grid iterations and the effectiveness of the Toeplitz approximation iterative algorithm, the LPC parameters and their corresponding sum of squared errors have almost converged to the correct solution before any intergrid transfers are conducted. Therefore, very little correction is made and the results presented in Figure 4.7 look remarkably similar to those of the straight iterative method.

As in the straight iterative method, the LPC parameters converge quickly and the sum of squared errors, S, decreases just as rapidly. Again, the synthesized speech (shown in solid lines) at the first fine grid iteration is unintelligible. But, by the sixth iteration, fairly good speech can be synthesized from the LPC parameters.

In this case, the *V-cycle* showed no improvements over the straight iterative algorithm used. This is partly due to the effectiveness of the iterative algorithm used and partly because three fine grid iterations were done before any intergrid transfers took place. This does illustrate one important point, however. If an iterative algorithm converges, adding the multigrid technique will neither slow down that convergence nor cause the algorithm to diverge.

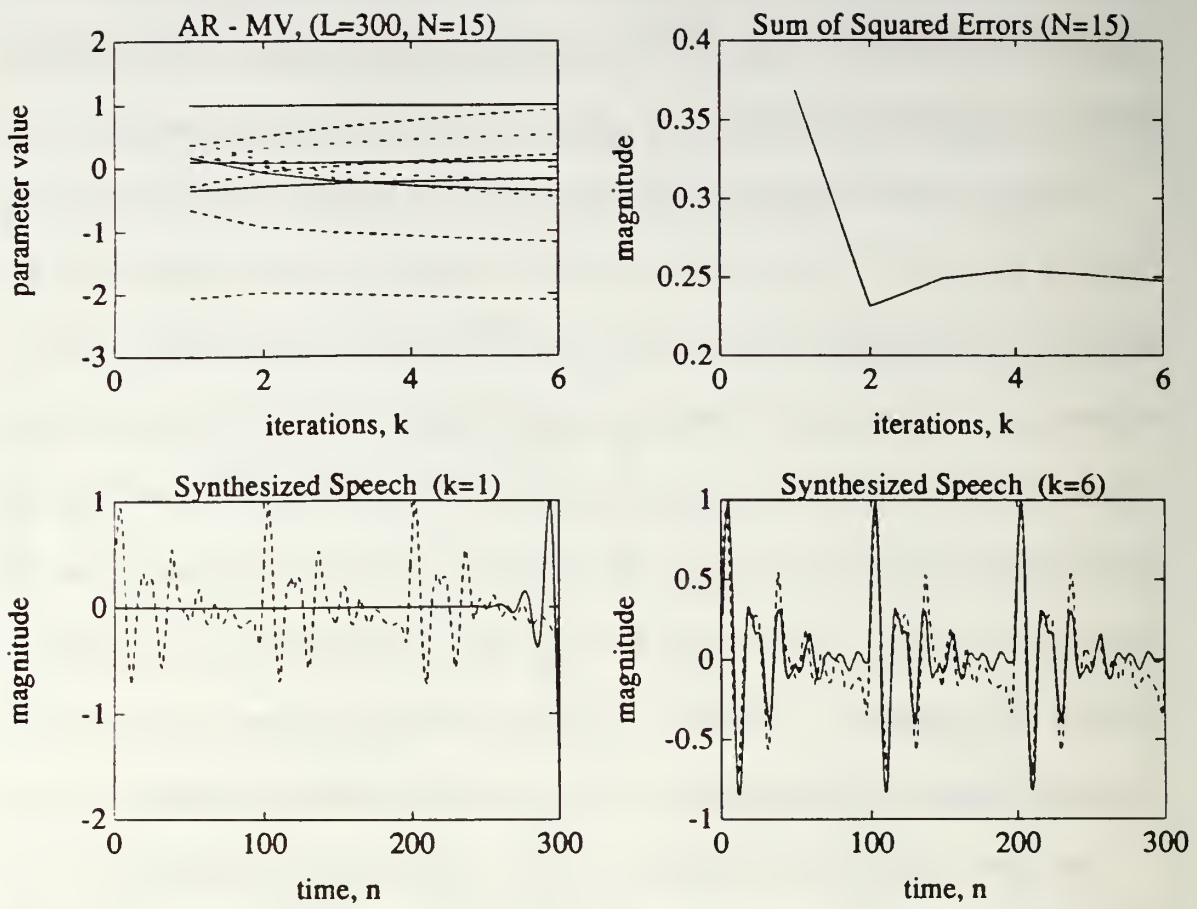

Figure 4.7: 15th Order AR Model of Voiced Speech Using *V-cycle* Method. Original speech waveform shown in dashed lines, synthesized speech waveforms shown in solid lines for $L=300$ data points.

In the *FMV-cycle* method shown in Figure 4.8, a remarkable increase in effectiveness can be seen. The sum of squared errors curve decreases much more smoothly than before and the synthesized speech at the first iteration is not only intelligible, but it matches closely to that of the sixth iteration. This is a case when the *FMV-cycle* may converge to a satisfactory solution without even performing a single fine grid iteration.

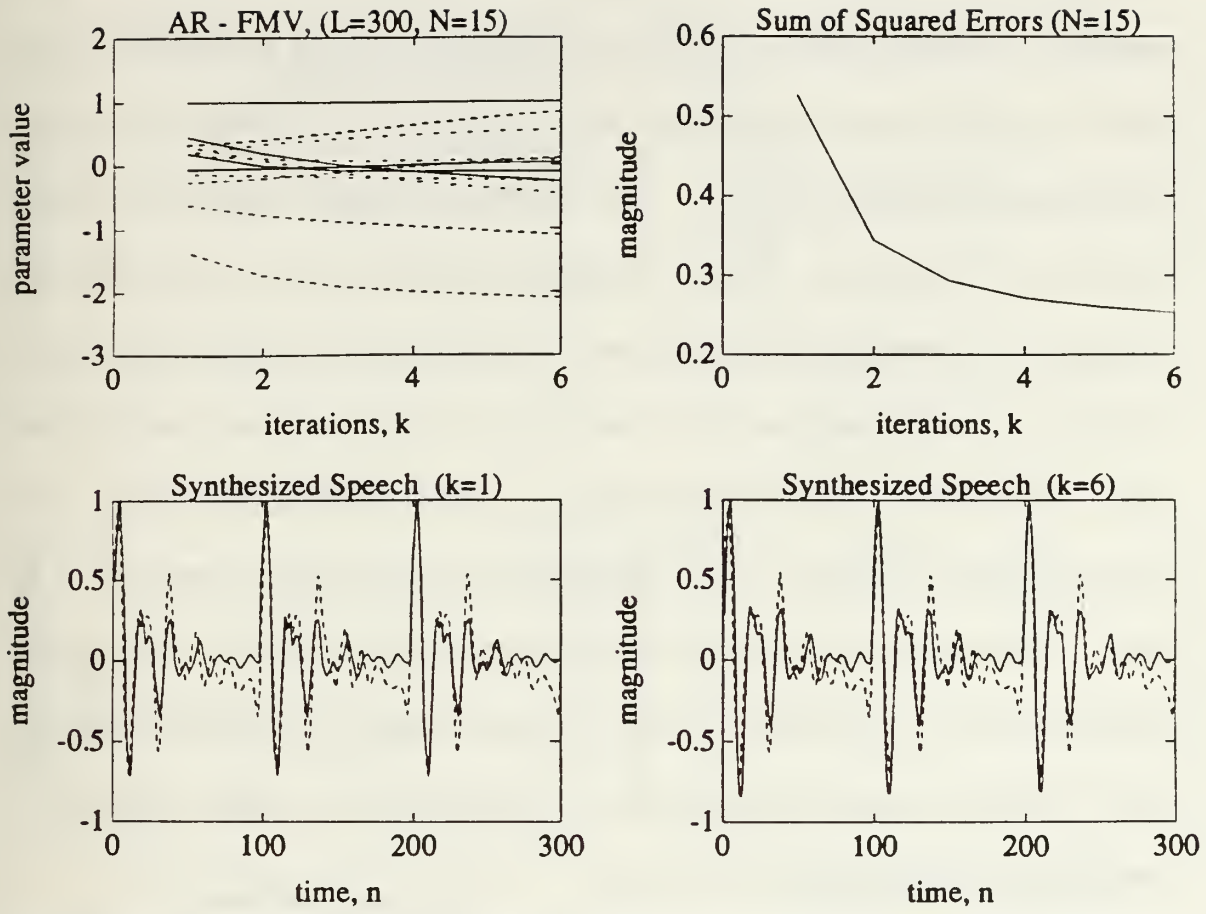

Figure 4.8: 15th Order AR Model of Voiced Speech Using the *FMV-cycle* Method. Original speech waveform shown in dashed lines, synthesized speech waveform shown in solid lines using $L=300$ data points.

C. IIR SIMULATIONS

The IIR simulations were conducted using the same 22nd order IIR elliptical bandpass filter used in the FIR simulation shown in Figure 4.1. As before, only one representative case is presented here. The results of the 31st order ($M=31$, $N=31$) IIR model simulation are discussed here and the additional simulation results covering model orders ranging from 7th to 63rd can be found in Appendix C (Figures C.1 - C.12).

A stability problem arose using the algorithm developed in Chapter III for IIR modeling. The problem occurred whenever an attempt was made to "over-model" the system, i.e., using a model order greater than the system order. The resulting frequency response was not affected, however. Only the impulse response of the model showed the effects of the instability. In order to compensate for this problem, a method of pole correction was developed. Using the ideas of a minimum phase filter in [Ref. 12], the algorithm checks for poles outside the unit circle. If poles are found outside the unit circle, leading to an unstable model, they are moved inside the unit circle to the conjugate reciprocal locations, thus maintaining the desired frequency response of the model. It should be noted that the instability problem was not due to the multigrid techniques used in the modeling process, but was inherent in the modeling algorithm used. The straight iterative algorithm had the same instability problem each time an attempt was made to over-model the system.

Figure 4.9 presents the results of the straight iterative method using $L=1000$ data points. The change in the filter parameters is most noticeable in the first two or three fine grid iterations and it seems that the parameters have converged by the sixth iteration. The filter impulse response is provided for comparison purposes. The frequency response of the resulting filter at the first iteration is remarkably recognizable and is very close to its true values by the sixth iteration.

The *V-cycle* method shown in Figure 4.10 shows little improvement over the straight iterative method for all the same reasons discussed in the section on AR simulations. With three iterations at each grid level, i.e. $\upsilon=3$, the IIR filter parameters

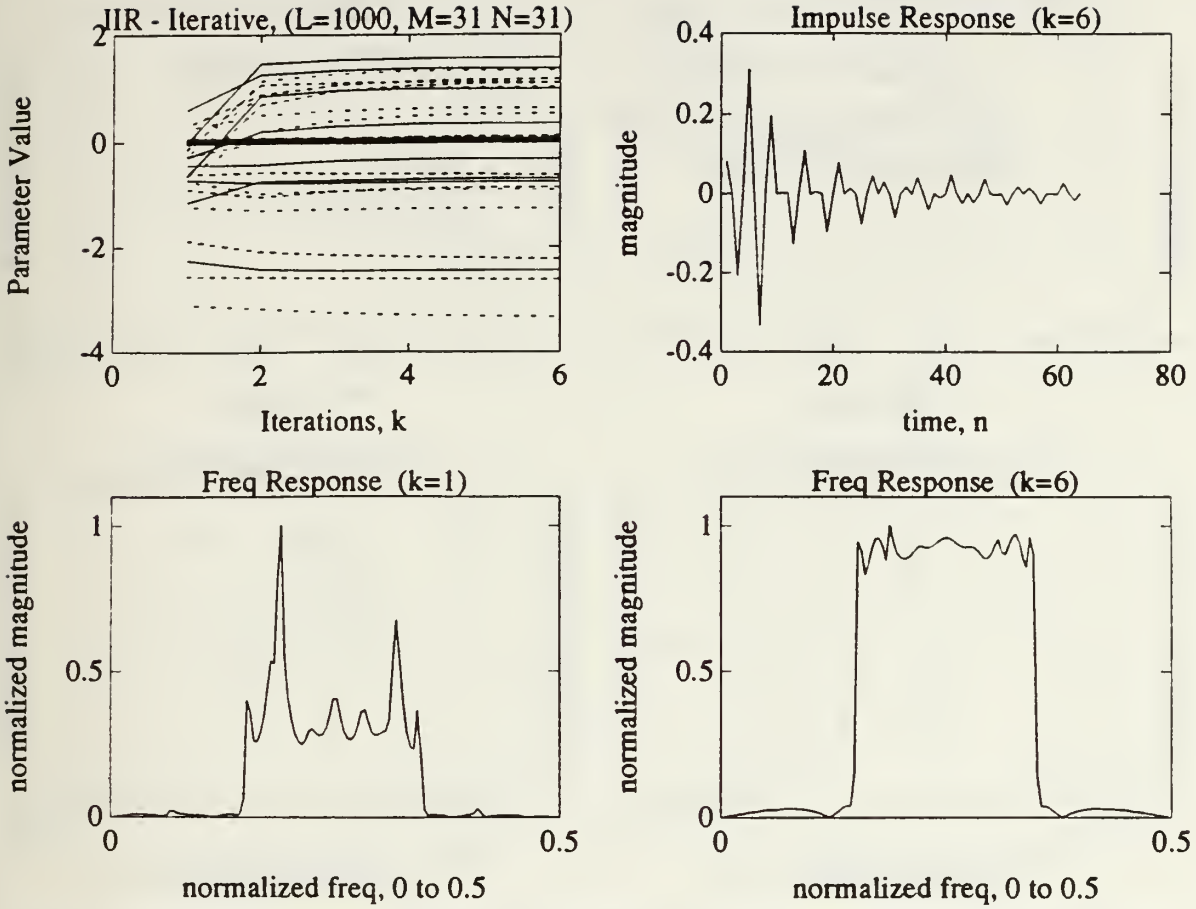

Figure 4.9: 31st Order IIR Model Using Straight Iterative Method. Impulse and frequency responses are shown using $L=1000$ data points.

nearly converged to their final solution prior to any intergrid transfers. Therefore, in this case, the benefits of the multigrid *V-cycle* are not fully realized.

Figure 4.11 presents the *FMV-cycle* simulation results. The fine grid iterations beyond the first two seem to be unnecessary and the impulse response at the sixth iteration is quite similar to the other two methods. The normalized frequency response at the first fine grid iteration is much improved over the straight iterative and V-cycle methods. This shows the effectiveness of the coarse grid correction and the nested

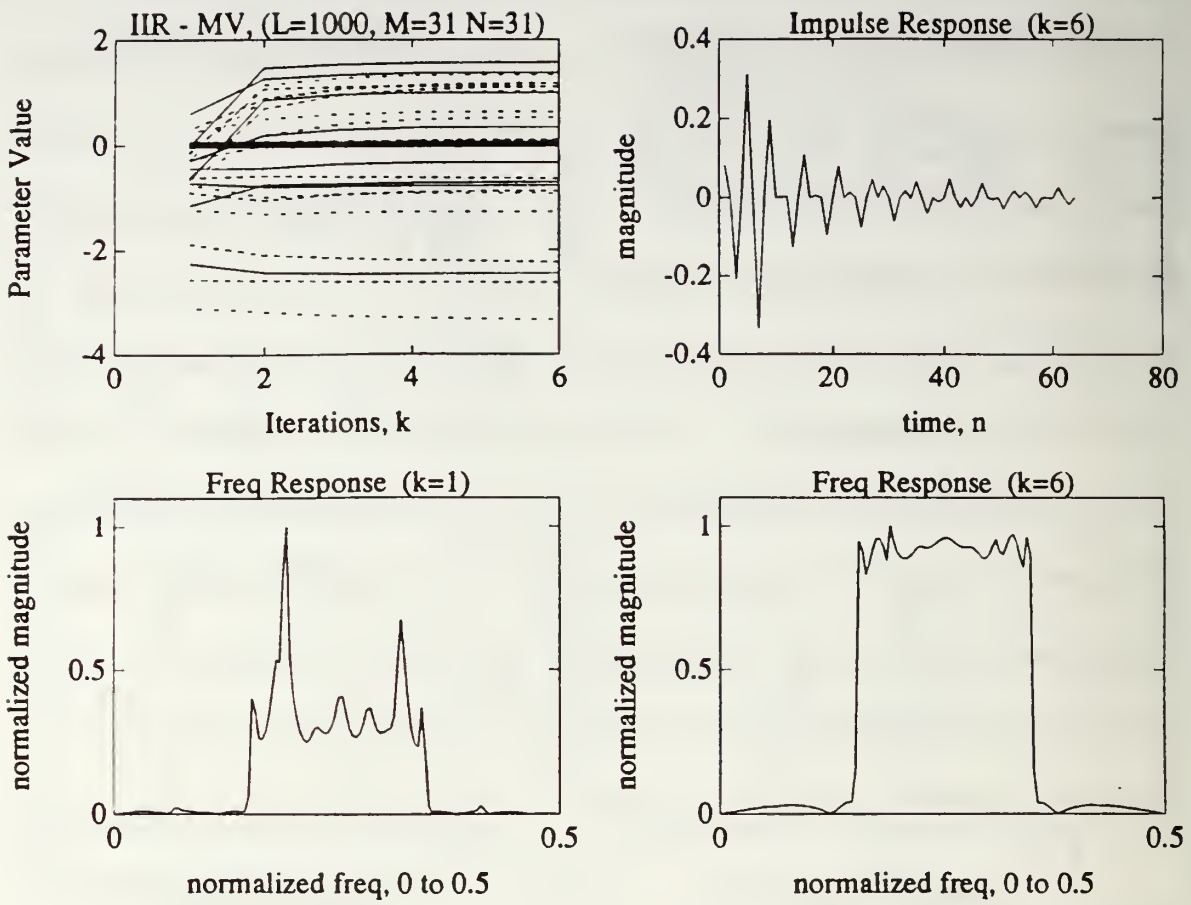

Figure 4.10: 31st Order IIR Model Using the *V-cycle* Method. Impulse and normalized frequency responses are shown using $L=1000$ data points.

iterations prior to the intergrid transfer to the finest grid. Finally, the frequency response at the sixth iteration is much closer to the system frequency response provided by the other methods. This is apparently because the low-frequency components of the error have been effectively removed prior to the transfer to the finest grid.

The impulse response error of the three methods at the sixth fine grid iteration is compared to the true impulse response of the system in Figure 4.12. A close scrutiny of each impulse response error reveals similar deviation from the actual impulse response

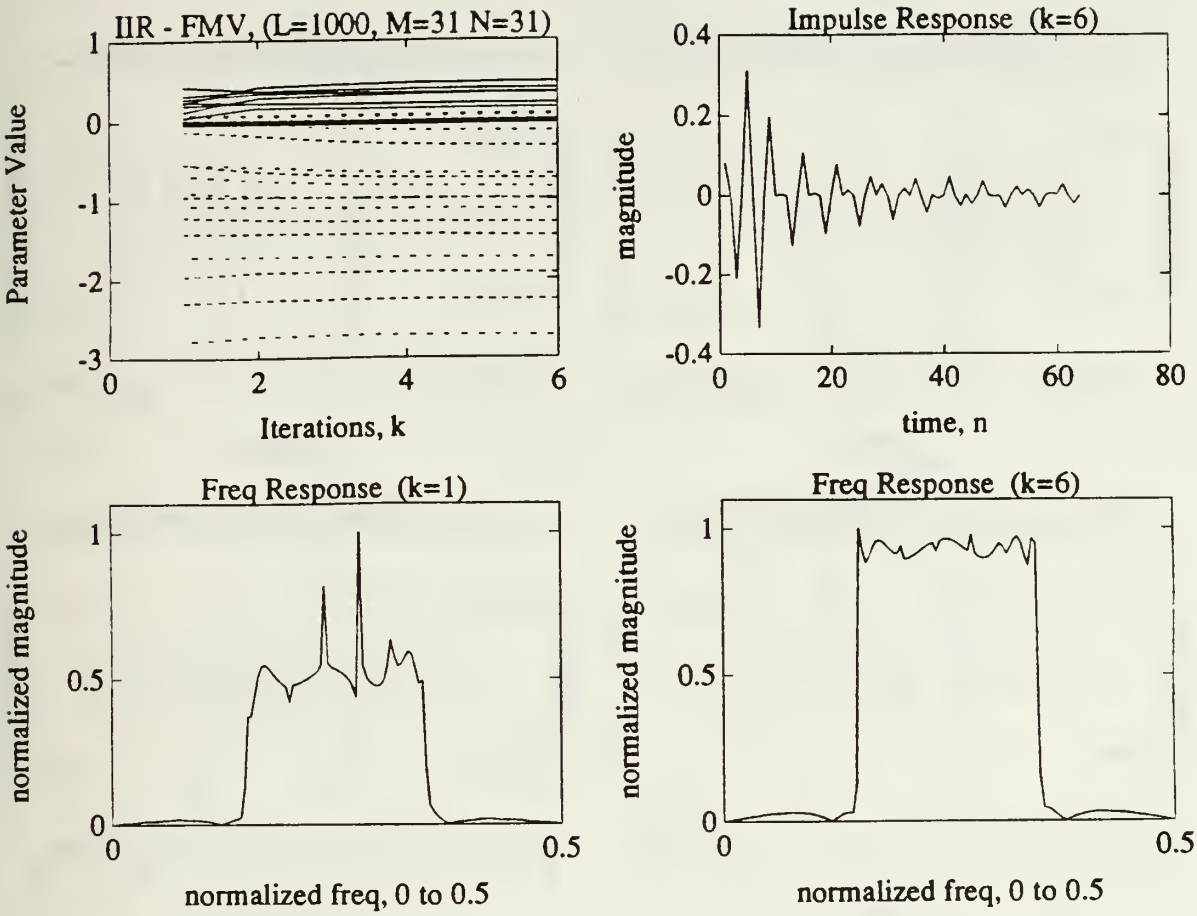

Figure 4.11: 31st Order IIR Model Using the *FMV-cycle* Method. Impulse and normalized frequency responses are shown using $L=1000$ data points.

although the *FMV-cycle* error is the smallest of the three methods. This supports the fact that multigrid techniques only enhance an iterative method and in no way degrade the results.

To conclude the discussion of the IIR simulations, we compare the phase responses of the systems, i.e., the true phase response of the system is compared to the phase responses of the straight iterative method, the *V-cycle* method, and the *FMV-cycle* method at the sixth fine grid iteration. From Figure 4.13 it is evident that there are some

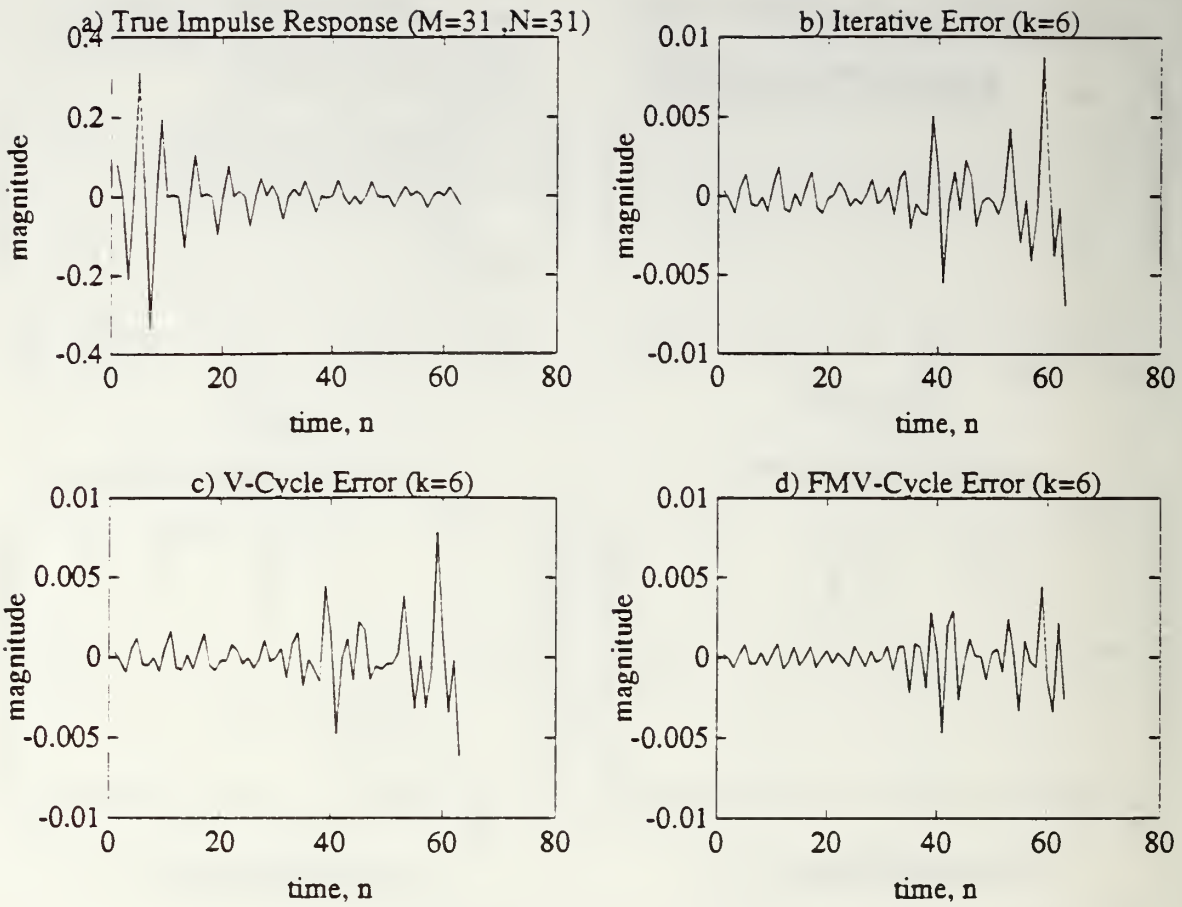

Figure 4.12: 31st Order IIR Model Impulse Response Error Comparison. a) True System Impulse Response; b) Iterative Method Error; c) *V-cycle* Method Error; d) *FMV-cycle* Method Error.

differences in the phase response of each method outside the passband (as there are in each of the frequency responses), however, the phase responses in the passband of each filter are quite similar. Since the magnitude of the frequency response outside the passband is small, the error in the phase response is relatively insignificant.

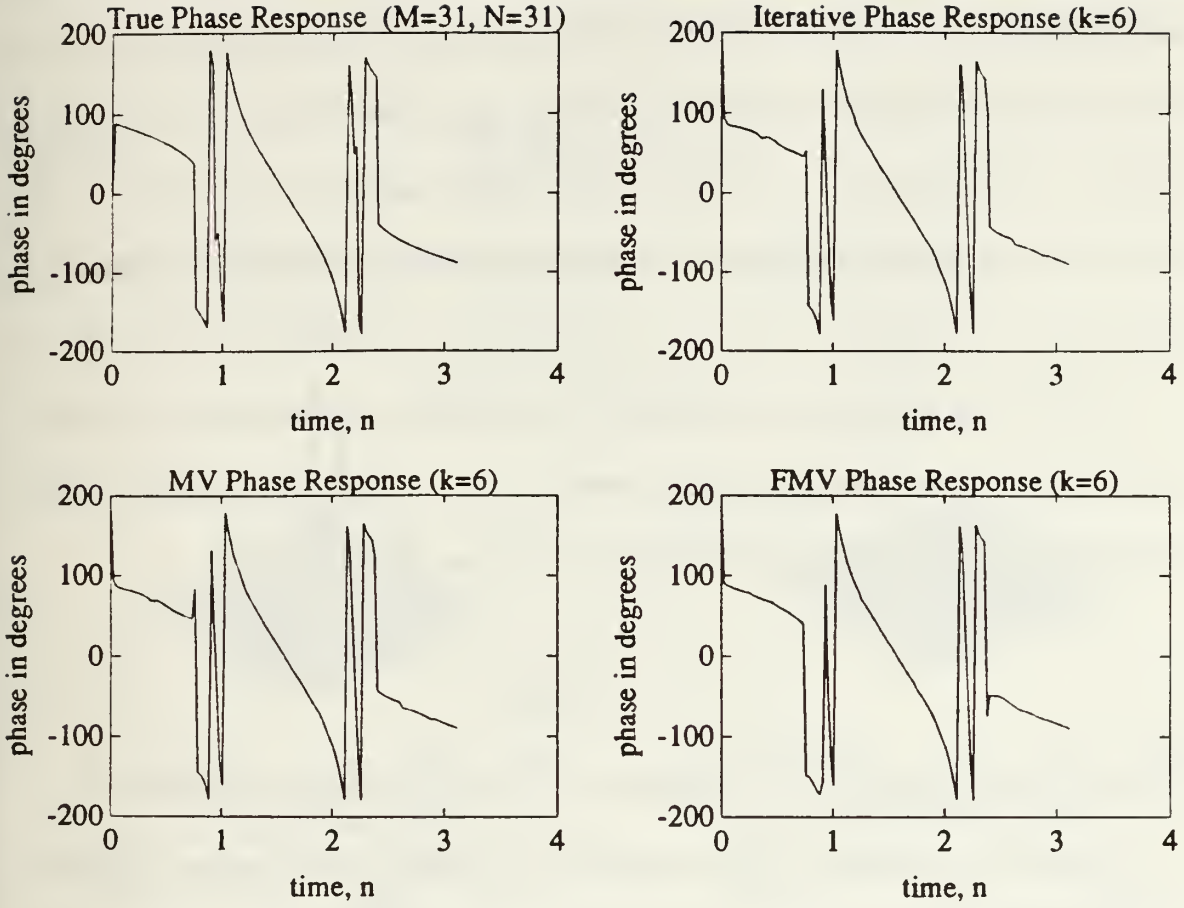

Figure 4.13: True vs. Modeled Phase Responses Using the Straight Iterative Method, *V-cycle* Method, and the *FMV-cycle* method.

D. 2-D AR SPECTRAL ESTIMATION SIMULATIONS

The 2-D AR modeling algorithm developed in Chapter III was used to estimate the parameters of a set of sinusoids in noise generated by

$$\begin{aligned} x(n_1,n_2) &= A_1\cos(2\pi f_{11}n_1 + 2\pi f_{12}n_2) \\ &+ A_2\cos(2\pi f_{21}n_1 + 2\pi f_{22}n_2) + \cdots \\ &+ A_p\cos(2\pi f_{p1}n_1 + 2\pi f_{p2}n_2) + w(n_1,n_2) \end{aligned} \qquad (4.1)$$

where the amplitudes A_{ij} are all equal to $\sqrt{2}$ and the f_{ij} are the normalized frequencies in the range $0 \leq f_{ij} \leq 0.5$. The term $w(n_1,n_2)$ is zero mean gaussian noise with variance,

σ_w^2. The value of σ_w^2 is chosen to give a desired signal-to-noise ratio (SNR) according to [Ref. 13]

$$SNR = 10\log_{10}\sum_{i=1}^{p} \frac{A_i^2}{2\sigma_w^2} \quad . \tag{4.2}$$

For all 2-D AR spectral estimation simulations conducted the SNR is maintained at 10 dB.

When using quarter-plane (QP) support, a method exists that involves combining the first and second quadrant spectral estimates [Ref. 6,8]. This method is called combined quadrant (CQ) [Ref. 14] support and is given by

$$\hat{P}_x = \frac{2\sigma_w^2}{[\hat{P}_{q1}^{-1} + \hat{P}_{q2}^{-1}]} \tag{4.3}$$

where $\boldsymbol{P}_{q1}$ is the first quadrant estimate and $\boldsymbol{P}_{q2}$ is the second quadrant estimate.

Numerous simulations were conducted with the 2-D AR spectral estimation problem. Filter mask sizes ranged from (3×3) to (9×9) with 2-D data sequence sizes ranging from (21×21) to (64×64) respectively. Along with these distinct changes, the sinusoidal frequencies were varied widely in order to determine whether or not the algorithm was spatially dependent. It was observed that the algorithm does provide different results as the spatial location of the frequencies is varied.

In this section, only one representative case is presented. A (36×36) data sequence with eight sinusoids in noise (SNR = 10 dB) with frequencies equally distributed in the frequency domain was modeled with the algorithm using k=5 fine grid iterations. The spectral estimate was then computed from the model parameters. The normalized

frequencies of the sinusoids used in this simulation were placed at (ω_1,ω_2) values of (0.1,0.1), (0.4,0.1), (0.25,0.17), (0.1,0.25), (0.4,0.25), (0.25,0.32), (0.1,0.4), and (0.4,0.4). The results of the straight iterative algorithm are displayed in Figures 4.14 and 4.15. The crosses indicate the true spatial frequencies. See Figures D.1 - D.18 in Appendix D to examine additional simulations.

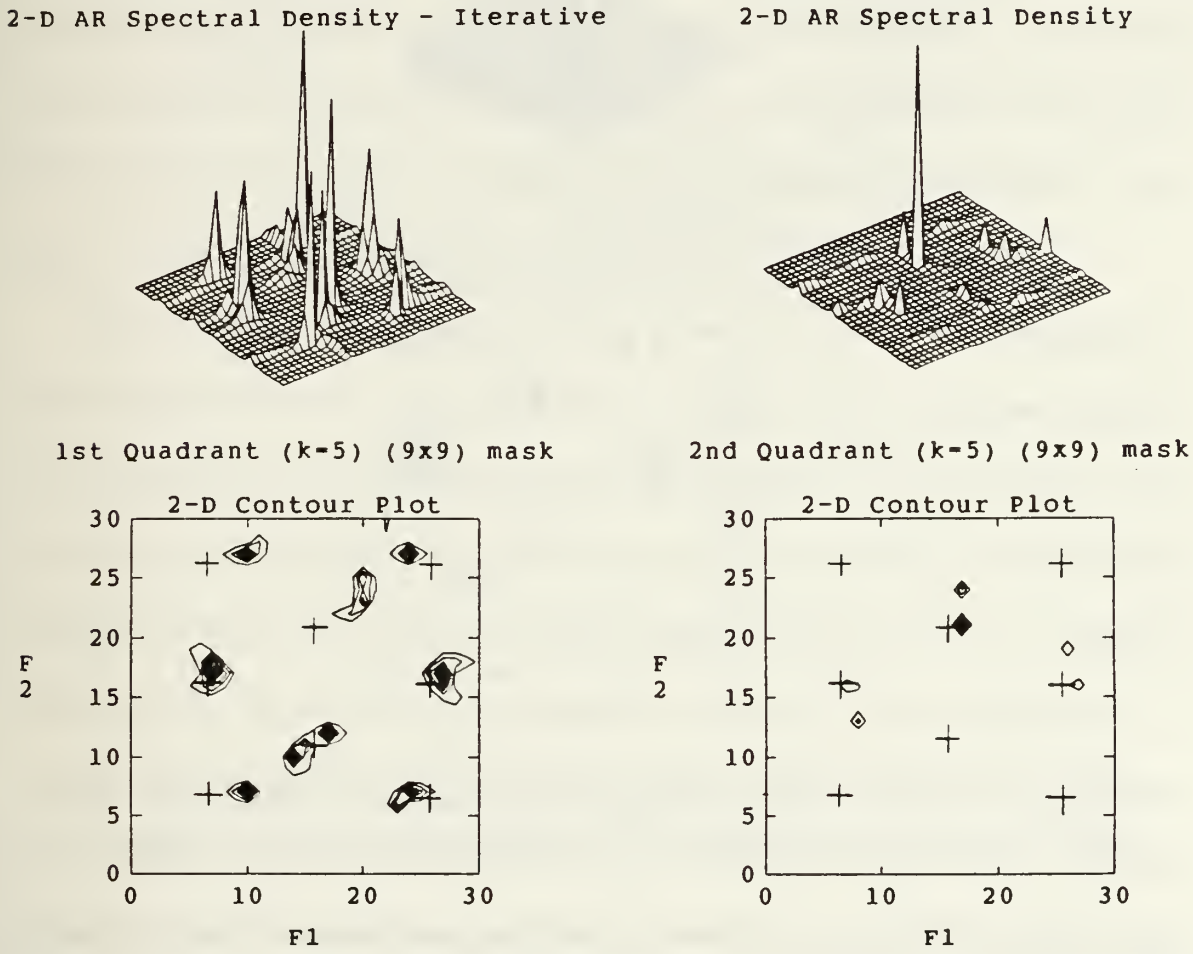

Figure 4.14: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR=10dB) Using the Straight Iterative Method. 1st & 2nd quadrant estimates are shown, (k=5, 9×9 mask).

The first quadrant estimation discerns all of the eight frequencies present with noticeable bias. The bias is thought to have arisen due to the ineffective means by which

most iterative algorithms remove the low-frequency components of the error [Ref. 7].

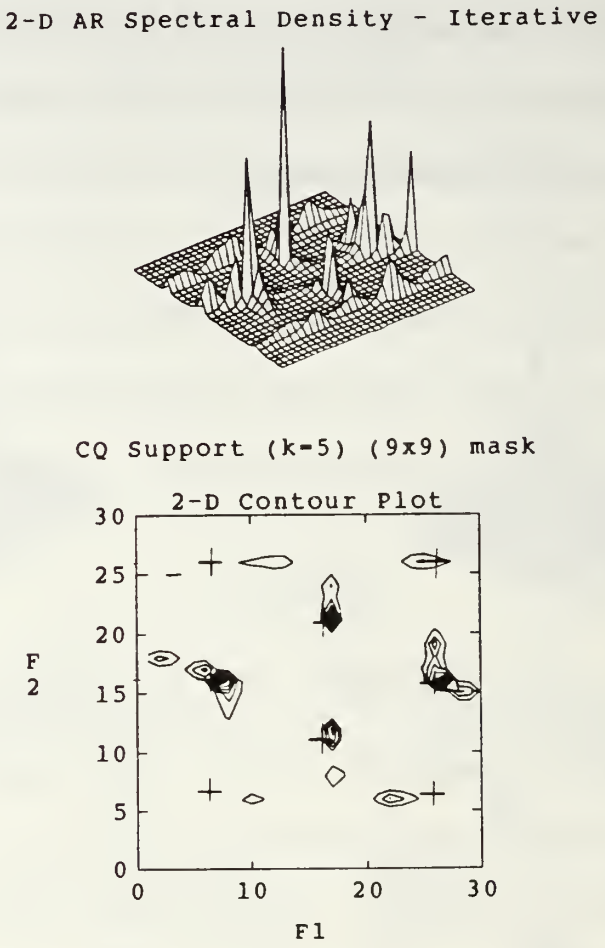

Figure 4.15: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR=10dB) Using the Straight Iterative Method. CQ estimate shown, (k=5, 9×9 mask).

The second quadrant estimate does not improve the spectral estimate. The first quadrant estimate completely dominates the second quadrant in that the combined quadrant estimate is affected mostly by the first quadrant estimate results. The combined quadrant estimate determines that eight sinusoids are present in the signal but with the noticeable bias of the first quadrant estimate.

It should be mentioned that as the number of iterations is increased the algorithm appears to diverge and the estimates degrade. This observation is similar to that reported in [Ref. 6]. In this case, the straight iterative method's performance degraded when ten fine grid iterations were conducted.

Figures 4.16 and 4.17 show the results of using the *V-cycle* method to first model and then estimate the spectral components of the system described above. In both the 1st and 2nd quadrant estimates, the average magnitude of the spectral peaks is much lower with very little detail appearing in either estimate. Not much about the nature of the signal, let alone the spatial frequencies, can be gathered from these results.

The *V-cycle* method CQ spectral estimate shown in Figure 4.17 demonstrates the power of the CQ method. Even when both the 1st and 2nd quadrant estimates appeared to be uninteresting, the CQ estimate displays great promise; eight sinusoids can be discerned. Most importantly, however, is that those frequencies shown are closer to the true spatial locations than those in the straight iterative method. This can be attributed to the removal of the low-frequency components of the error by the *V-cycle* method.

Like the *V-cycle* case, the *FMV-cycle* simulation in Figures 4.18 and 4.19 produces unrecognizable results in the 1st and 2nd quadrant estimates. The spectral peaks are irregular in terms of their magnitude and only a few of the correct spatial frequencies can be perceived from the contour plots. However, significant improvement results in the CQ method. Figure 4.19 shows the CQ spectral estimate using the *FMV-cycle* method. All eight sinusoids are placed close to their true spatial frequencies. Once again these superior results can be attributed to the removal of the low-frequency components of the

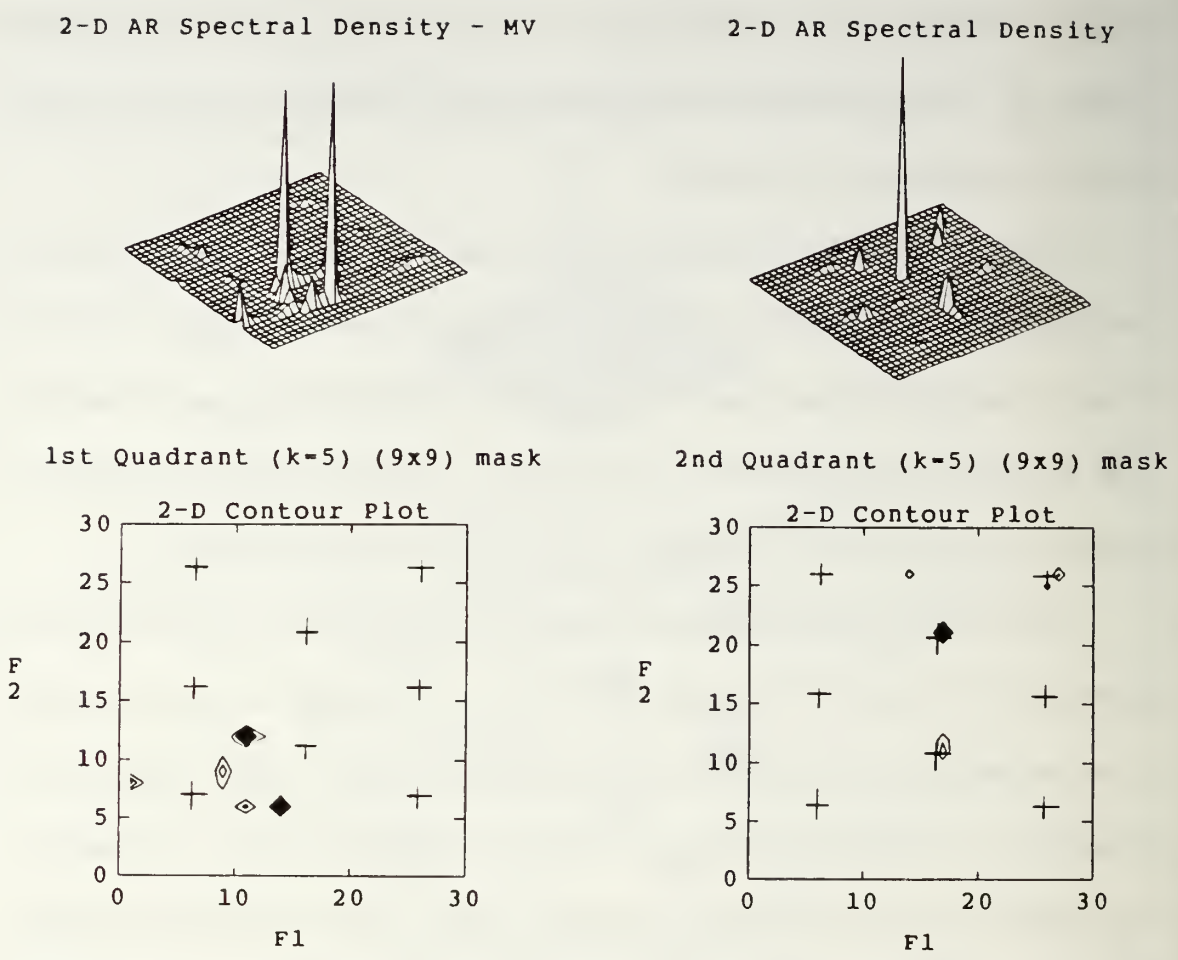

Figure 4.16: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR=10dB) Using the *V-cycle* Method. 1st and 2nd quadrant estimates shown, (k=5, 9×9 mask).

error by the combination of the favorable attributes of the coarse grid correction and nested iteration schemes in the *FMV-cycle* method.

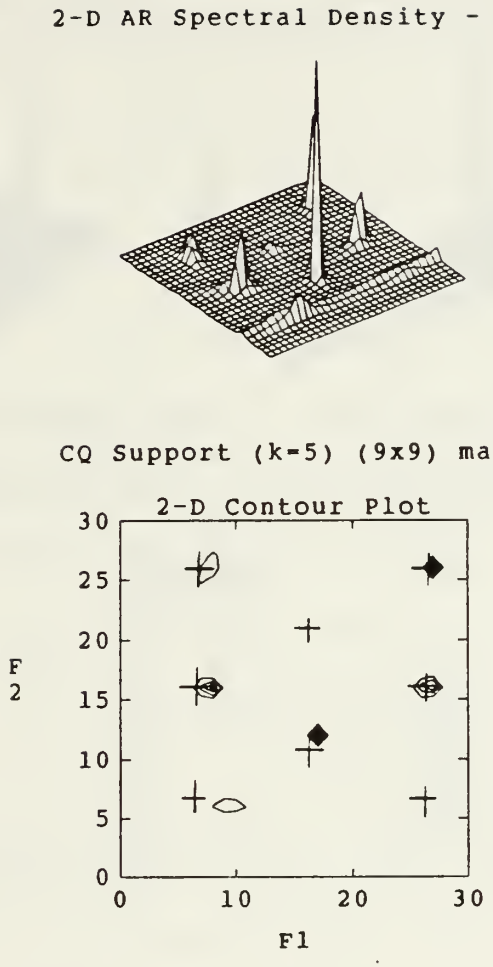

Figure 4.17: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR=10dB) Using the *V-cycle* Method. CQ estimate shown, (k=5, 9×9 mask).

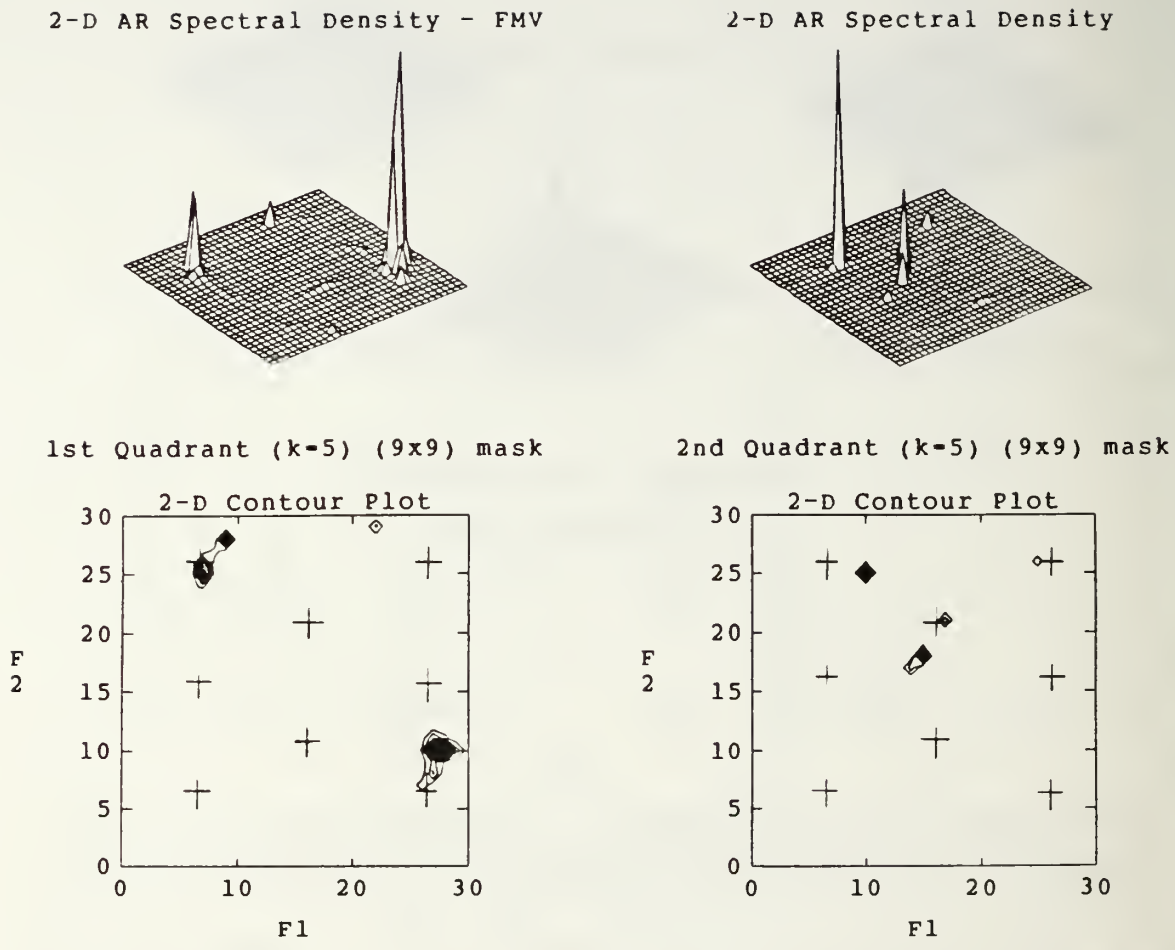

Figure 4.18: 2-D Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the *FMV-cycle* Method. 1st and 2nd quadrant estimates are shown, (k=5, 9×9 mask).

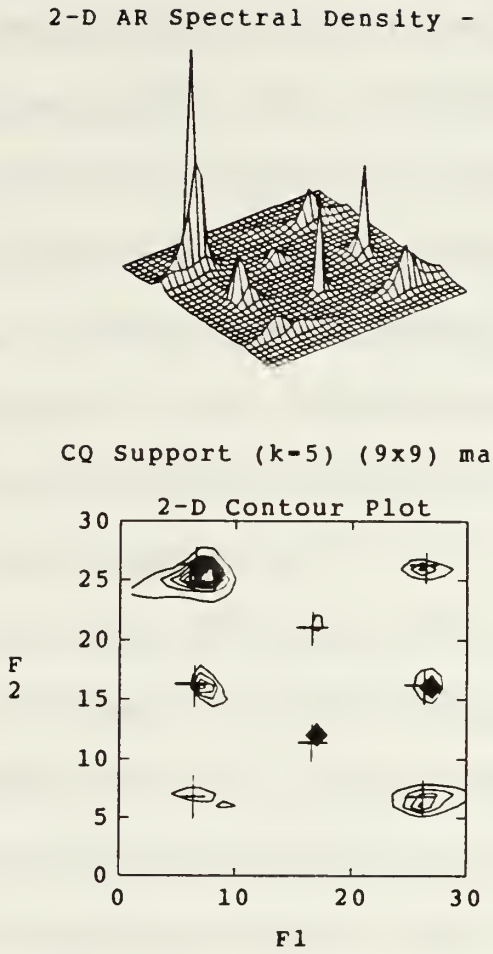

Figure 4.19: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the *FMV-cycle* Method. CQ estimate is shown, (k=5, 9×9 mask).

V. CONCLUSIONS

A. PERFORMANCE EVALUATION SUMMARY

In this thesis, the theory of multigrid is developed and applied to several system modeling applications. The results of these simulations are presented for representative cases in Chapter IV; many more are available for examination in the appendices. These results have all been quite favorable, but prior to drawing any conclusions, a discussion of the limitations of the multigrid techniques is in order.

Multigrid techniques are limited mainly by the intergrid transfer operators. Recalling from Chapter II, the linear interpolation operator transfers vectors from $\Re^{(N/2-1)}$ to $\Re^{(N-1)}$ space. For the case of $N=8$, the linear operator will transfer a vector of coefficients from three to seven. The limitation spoken of is that the vector length must always be odd. In the one-dimensional case, this means that only odd order FIR and AR models can take advantage of multigrid. For the IIR modeling case, the length of the coefficient vector must be odd which will determine the orders of the numerator and denominator polynomials used in modeling the particular system. Finally, for the 2-D AR modeling problem, the number of coefficients in the filter mask must be odd. This means that if a square filter mask is to be utilized the filter masks must be of sizes such as 3×3, 5×5, 7×7, etc. This is a limitation of the intergrid transfer operators, but is usually one that can be easily worked around.

The effectiveness of the *V-cycle* and the *FMV-cycle* is directly proportional to the order of the model used. As a general rule, the higher the order the greater the computational savings. Model orders or coefficient vector lengths on the finest grid of 2^n-1, where $n \geq 1$ is an integer, allow the multigrid cycles to provide the maximum advantage. For example, for the case of n=5, the *V-cycle* will progress through the following grids in a telescoping fashion: 31-15-7-3-1-3-7-15-31. Note that all of the grids visited by the *V-cycle* are odd and thus the cycle may progress to a theoretical single point and then telescope back up to the finest grid. As another example, take the coefficient vector length on the finest grid of 33. The first intergrid transfer to a coarser grid results in a vector length of 16, which is an even number. The *V-cycle* will then progress as follows: 33-16-33. This does not take advantage of what the coarse grid correction and nested iteration schemes can offer.

If the limitations of the intergrid transfer operators can be dealt with properly, the multigrid techniques offer significant advantages over the straight iterative methods. It has been shown that the use of the *V-cycle* and especially the *FMV-cycle* leads to more efficient fine grid iterations. In most applications the multigrid cycles are more effective and computationally less complex than the iterative method used alone. The actual savings realized depends on the specific application, the type of iterative method used, and most importantly the length of the coefficient vector. In order to achieve maximum benefit of the multigrid technique, the coefficient or parameter vector must be of length 2^n-1.

B. RECOMMENDATIONS FOR FUTURE WORK

The multigrid technique can be applied to solving any system of equations of the form $\boldsymbol{Ra} = \boldsymbol{r}$ in conjunction with an iterative method. This in itself leads to an endless list of possible applications that can benefit from the ideas presented here. The technique can be applied on multiple level problems, a variation not discussed here, which provides even more flexibility. [Ref. 7] The work reported here can be easily extended to data adaptive filtering. Multigrid techniques should also be explored for the possible application in the eigenvalue problem and the singular value decomposition (SVD) problem. Wherever large systems of equations of the form discussed in this thesis are being solved iteratively, the multigrid techniques should be evaluated for possible computational savings and/or an increase in the rate of convergence of the iterations.

APPENDIX A: FIR SIMULATIONS

This appendix provides further FIR modeling simulations of the reference system described in Chapter IV. Filter orders of 31, 63, and 255 are presented using a data sequence of $L = 1000$ points created by filtering zero mean, unit variance white noise with the bandpass filter of Figure 4.1. Graphical results of these filters are provided for the straight iterative, the *V-cycle*, and the *FMV-cycle* methods.

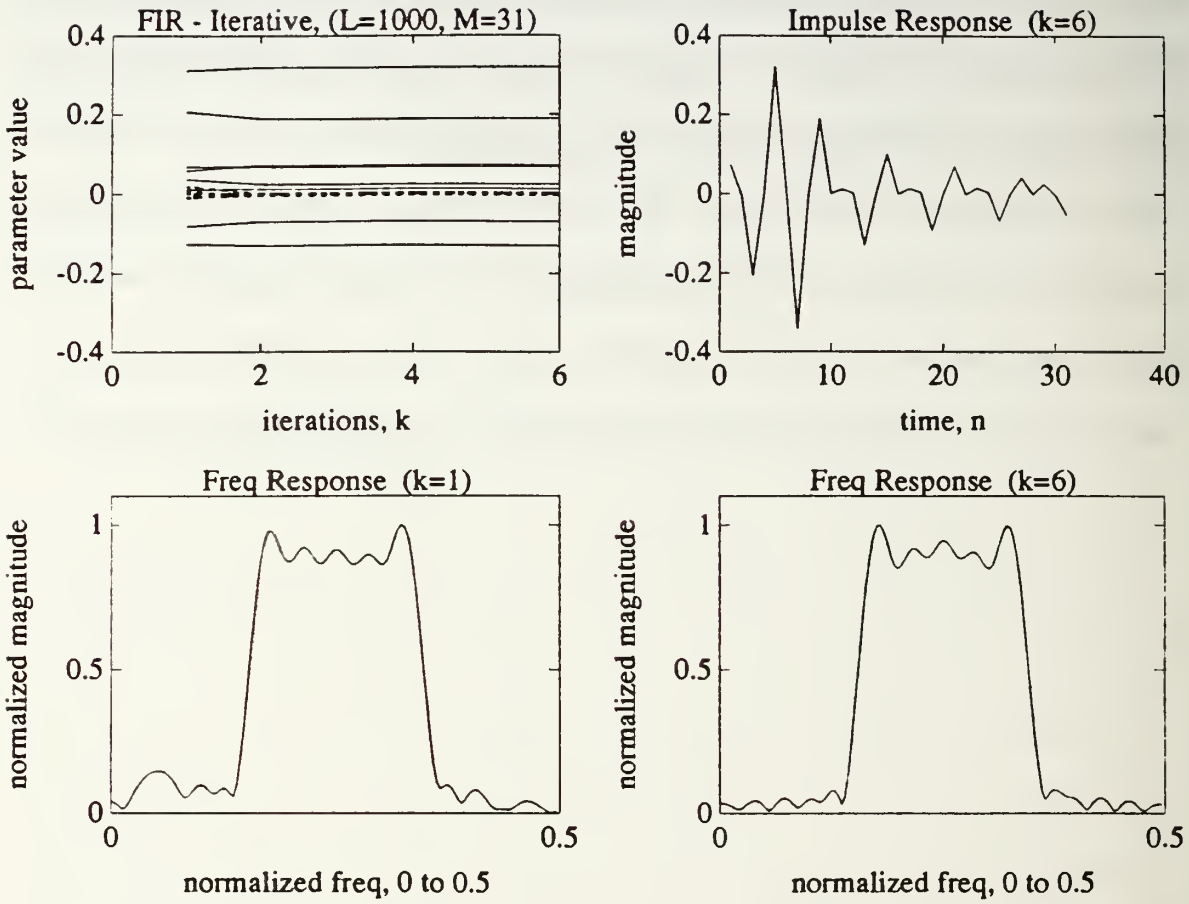

Figure A.1: 31st Order FIR Model Using the Straight Iterative Method. Impulse and normalized frequency responses are shown using $L=1000$ data points.

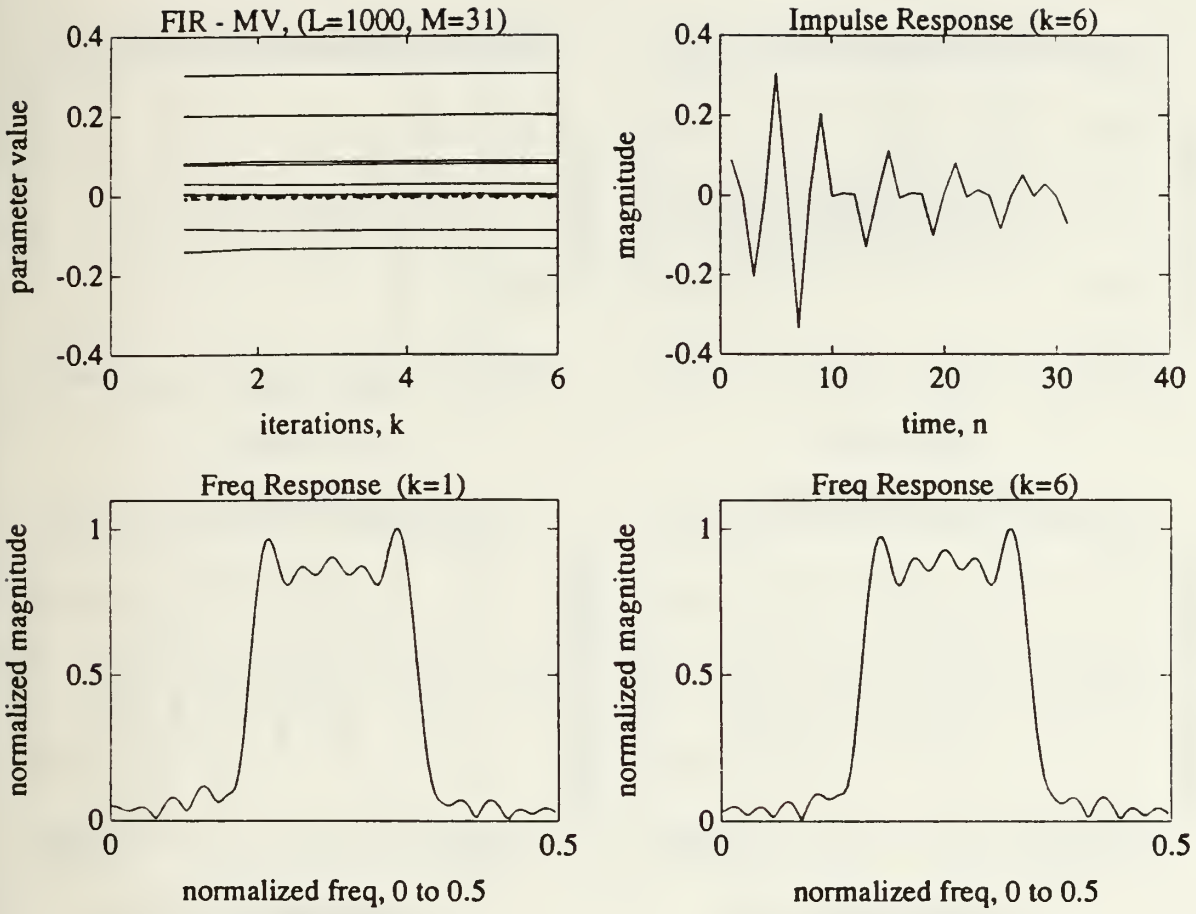

Figure A.2: 31st Order FIR Model Using the *V-cycle* Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

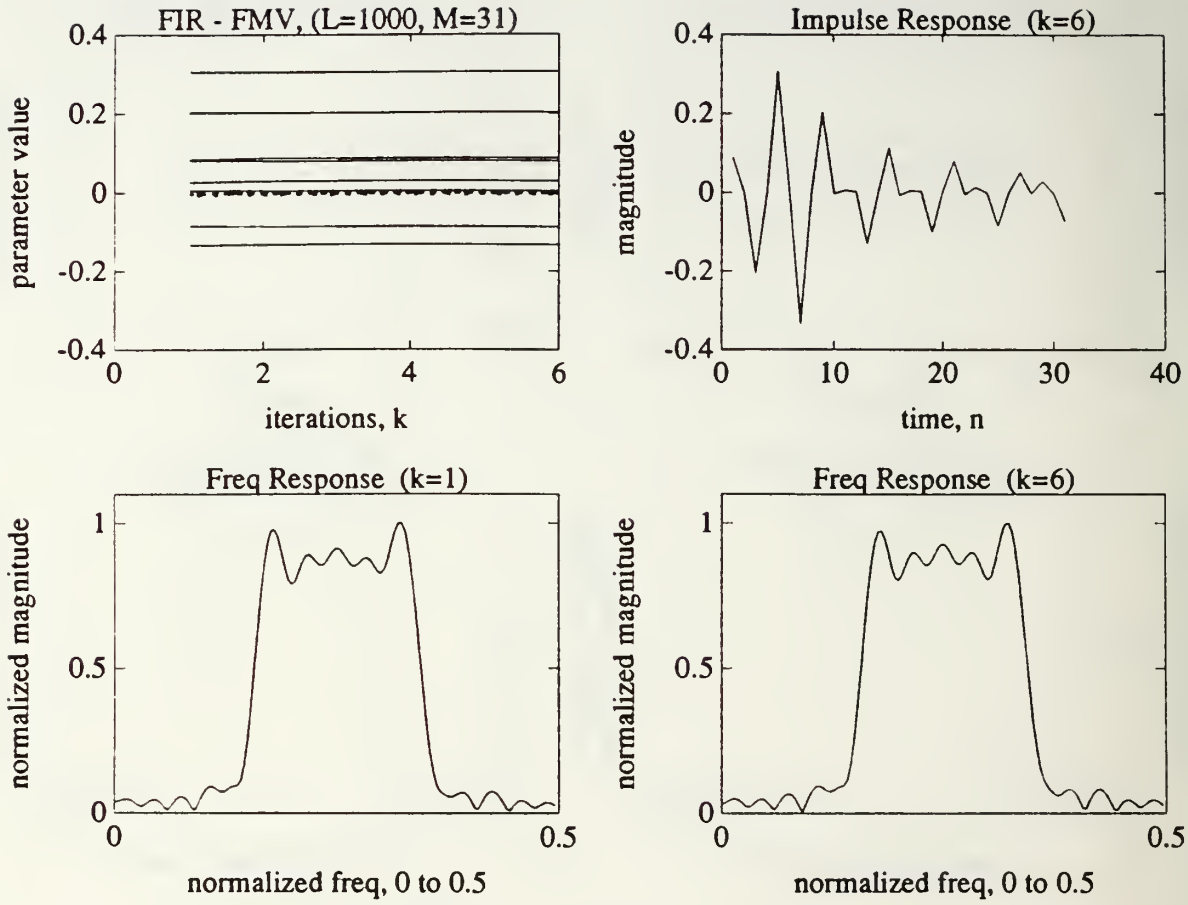

Figure A.3: 31st Order FIR Model Using the *FMV-cycle* Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

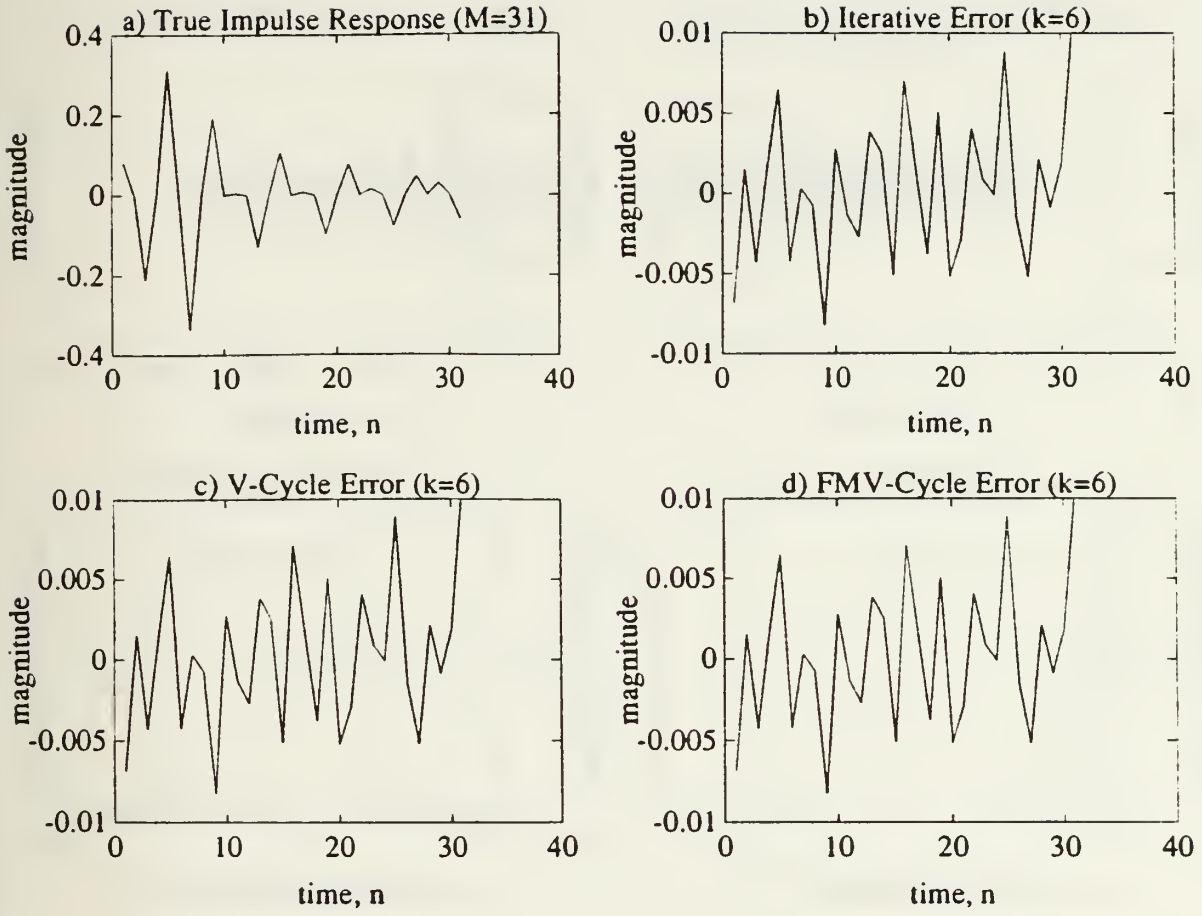

Figure A.4: 31st Order FIR Model Impulse Response Comparison. a) True System Impulse Response; b) Iterative Method Error; c) *V-cycle* Method Error; d) *FMV-cycle* Method Error.

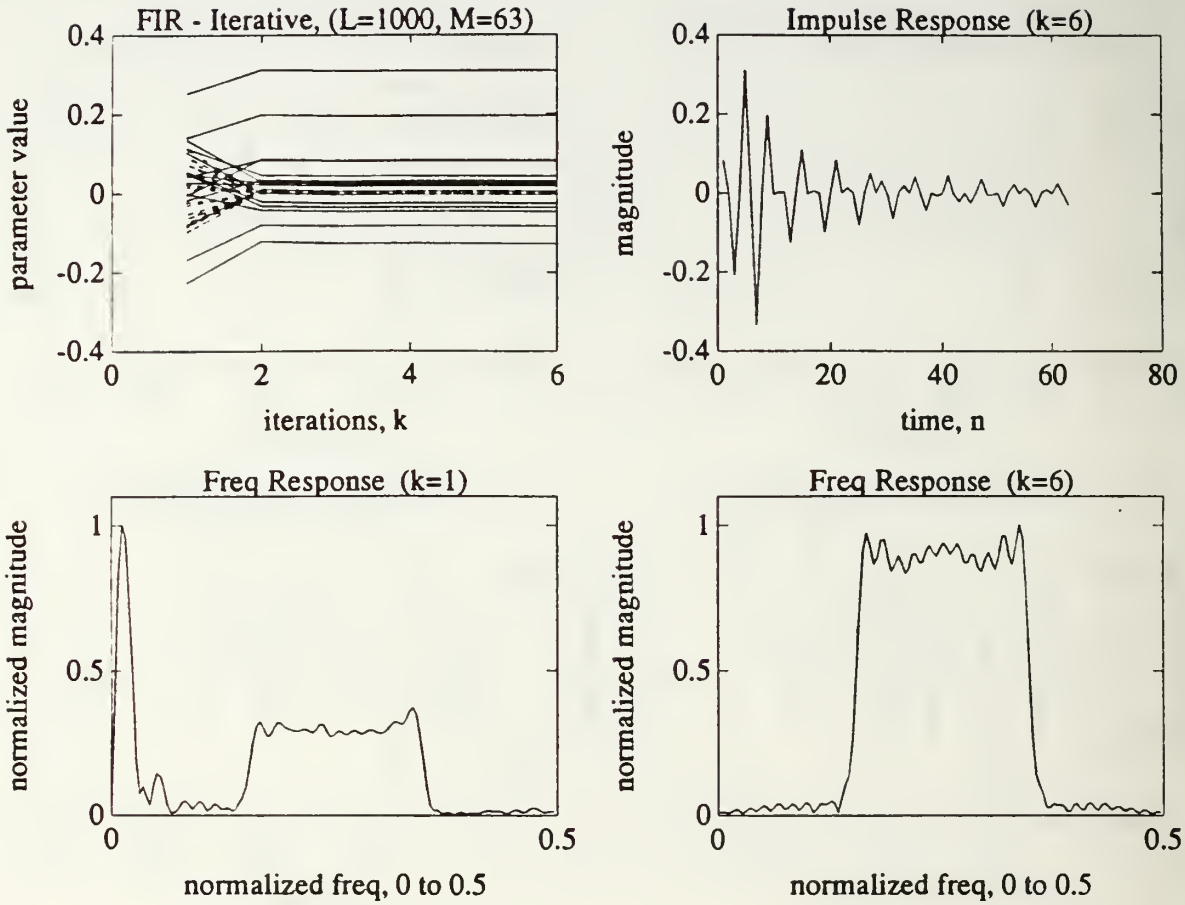

Figure A.5: 63rd Order FIR Model Using the Straight Iterative Method. Impulse and normalized frequency responses shown using $L = 1000$ data points.

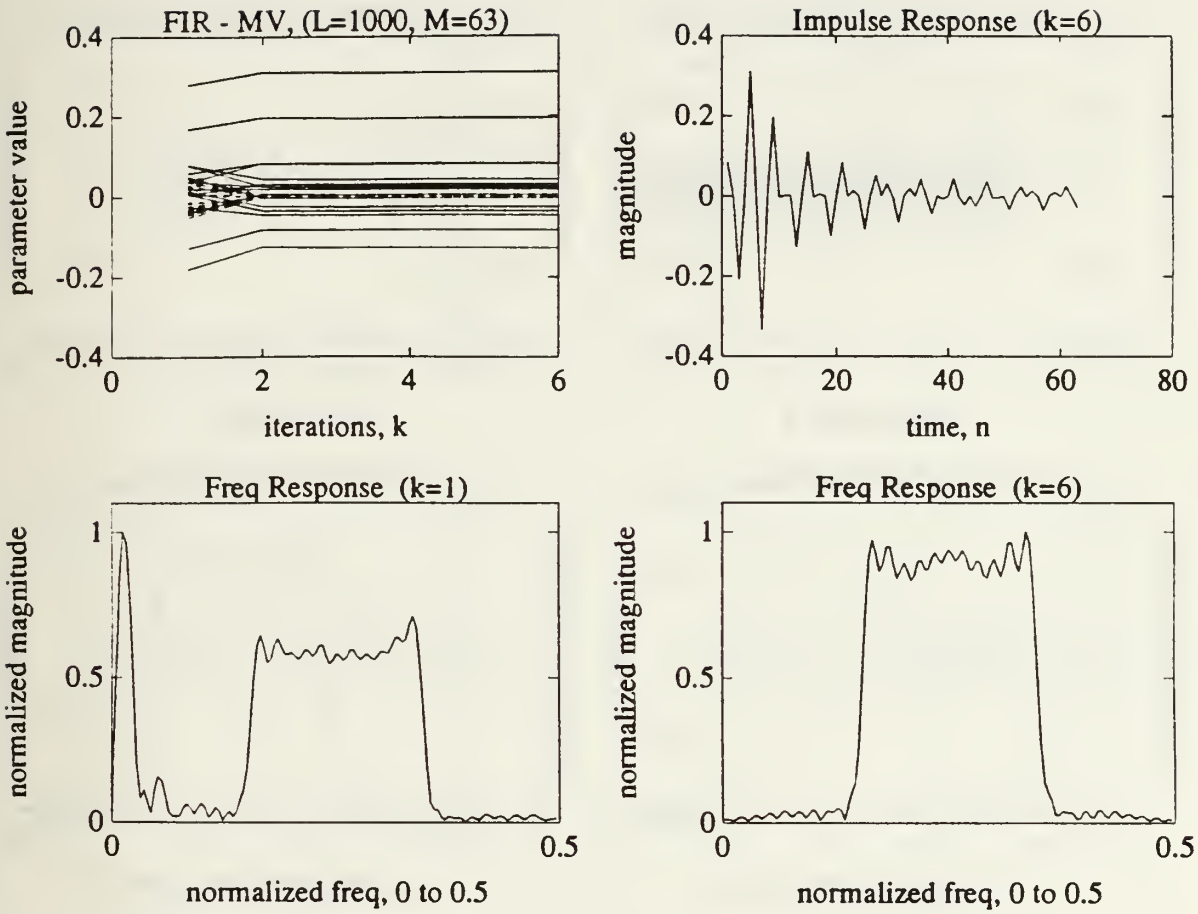

Figure A.6: 63rd Order FIR Model Using the *V-cycle* Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

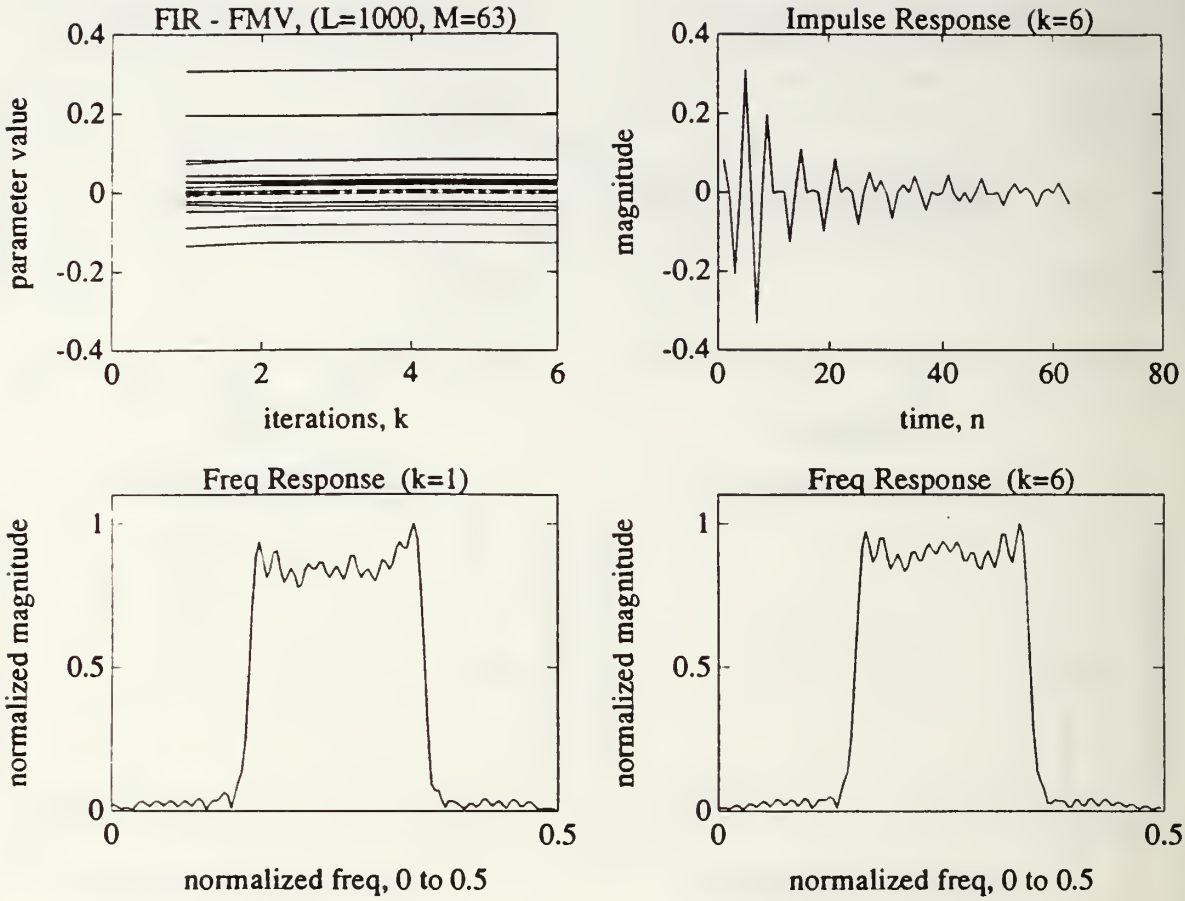

Figure A.7: 63rd Order FIR Model Using the *FMV-cycle* Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

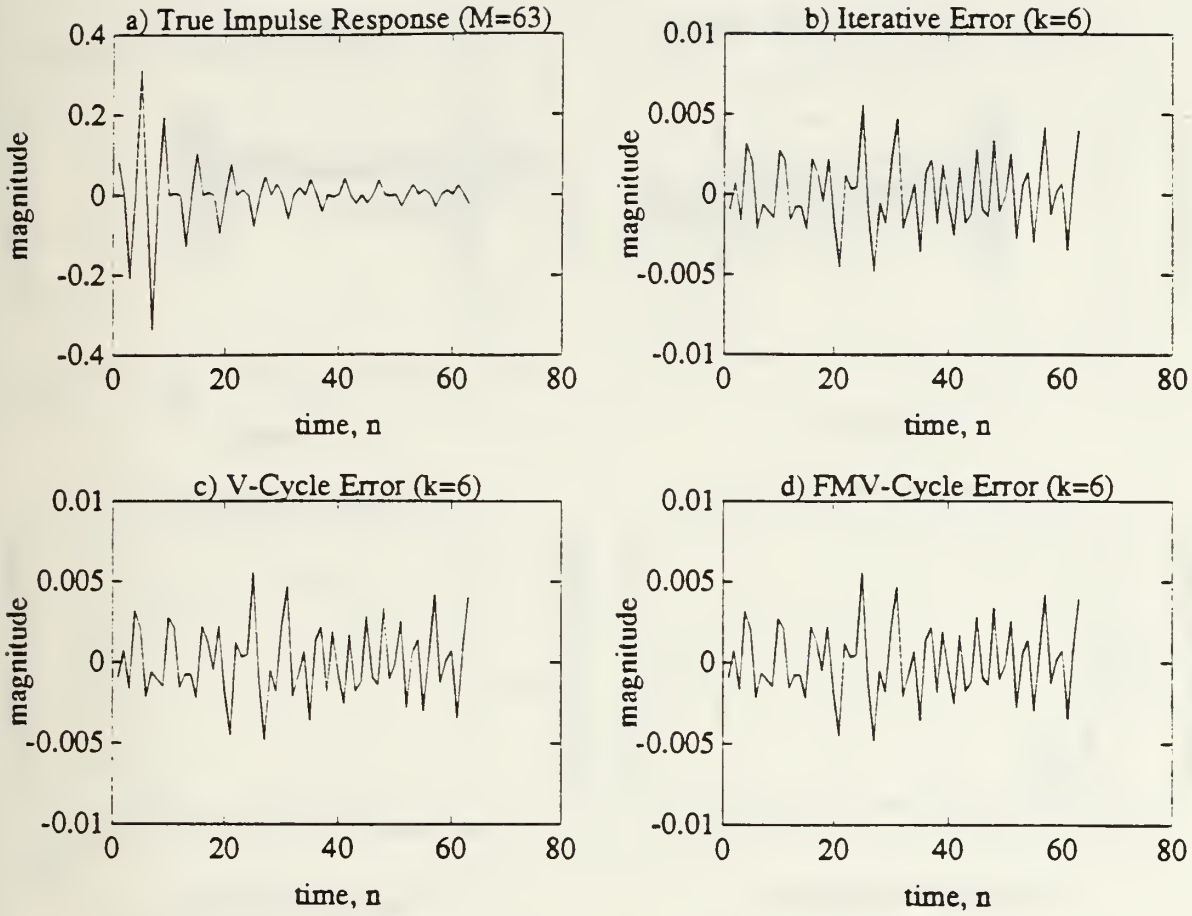

Figure A.8: 63rd Order FIR Impulse Response Error Comparison. a) True System Impulse Response; b) Iterative Method Error; c) *V-cycle* Method Error; d) *FMV-cycle* Method Error.

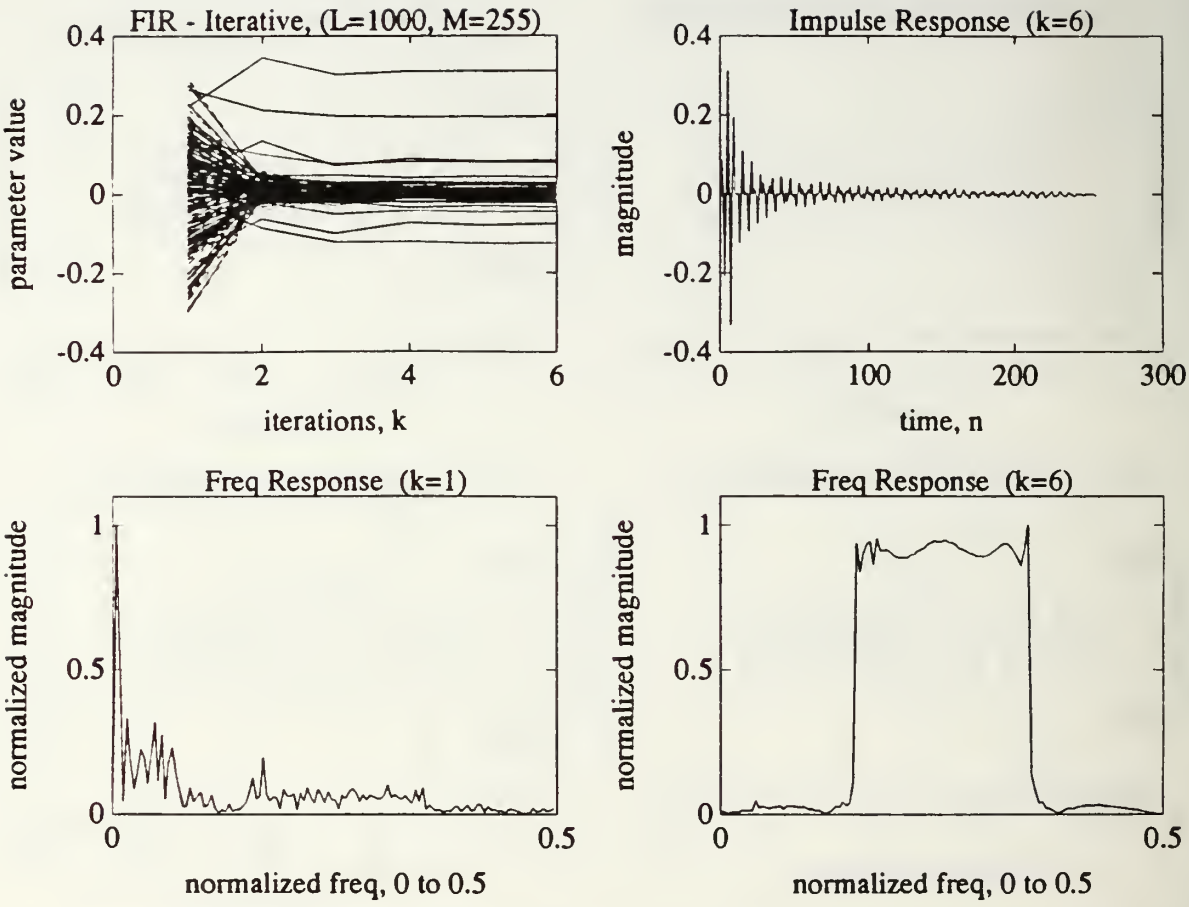

Figure A.9: 255th Order FIR Model Using the Straight Iterative Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

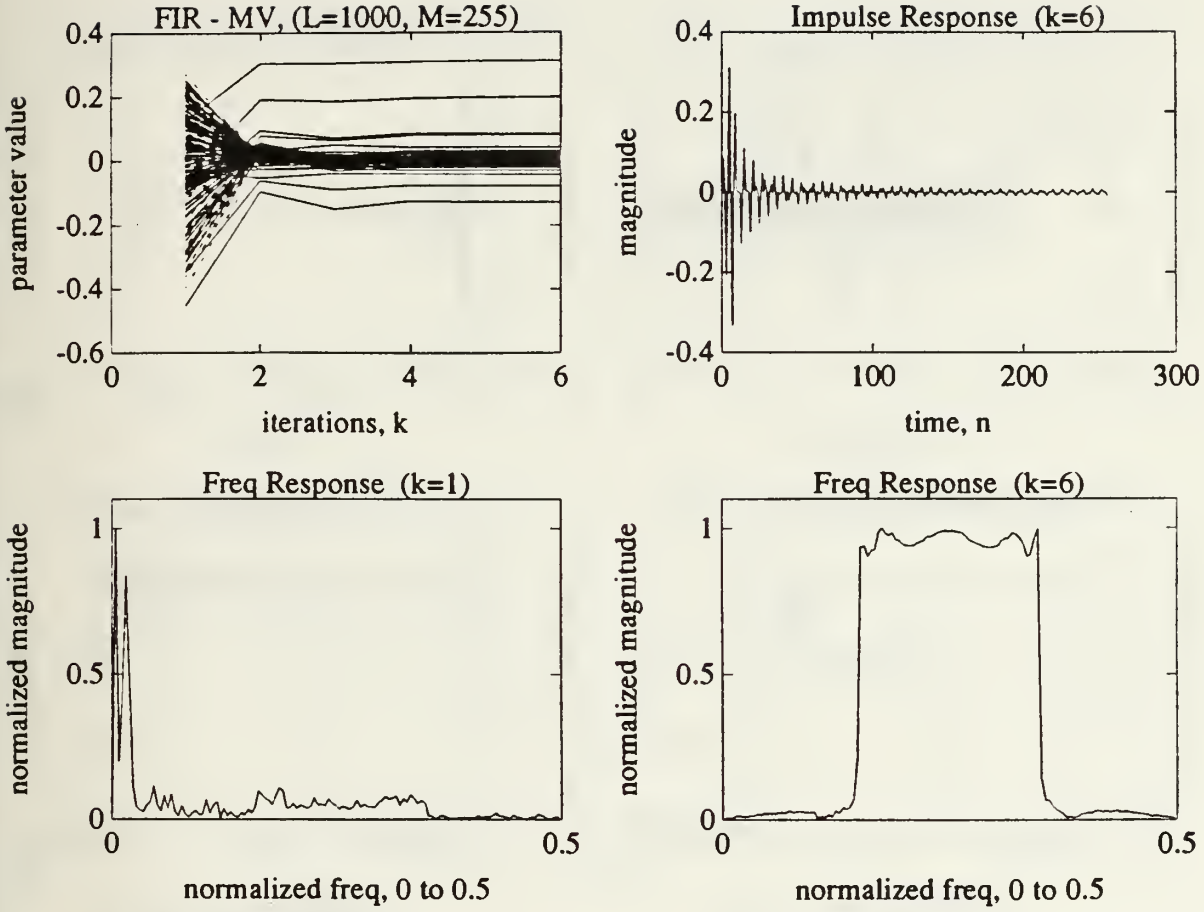

Figure A.10: 255th Order FIR Model Using the *V-cycle* Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

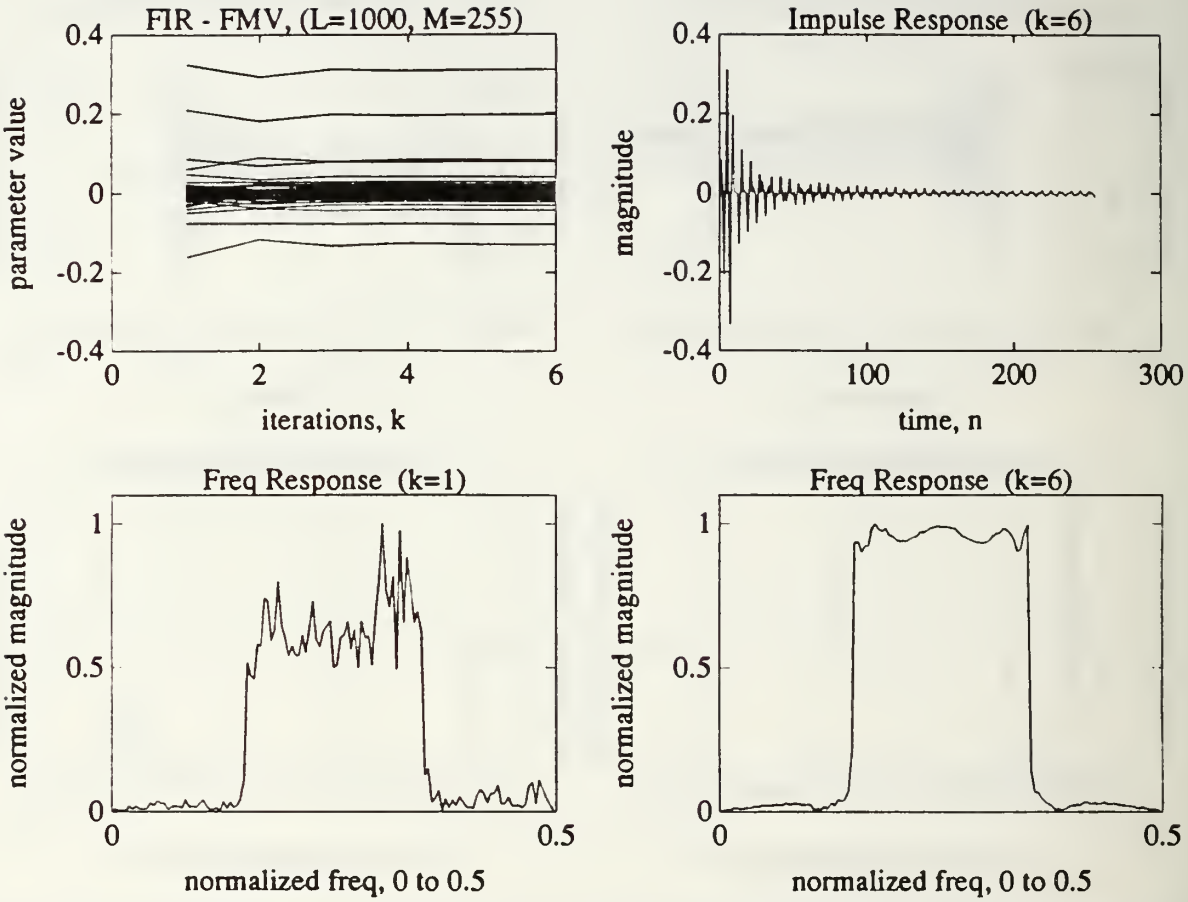

Figure A.11: 255th Order FIR Model Using the *FMV-cycle* Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

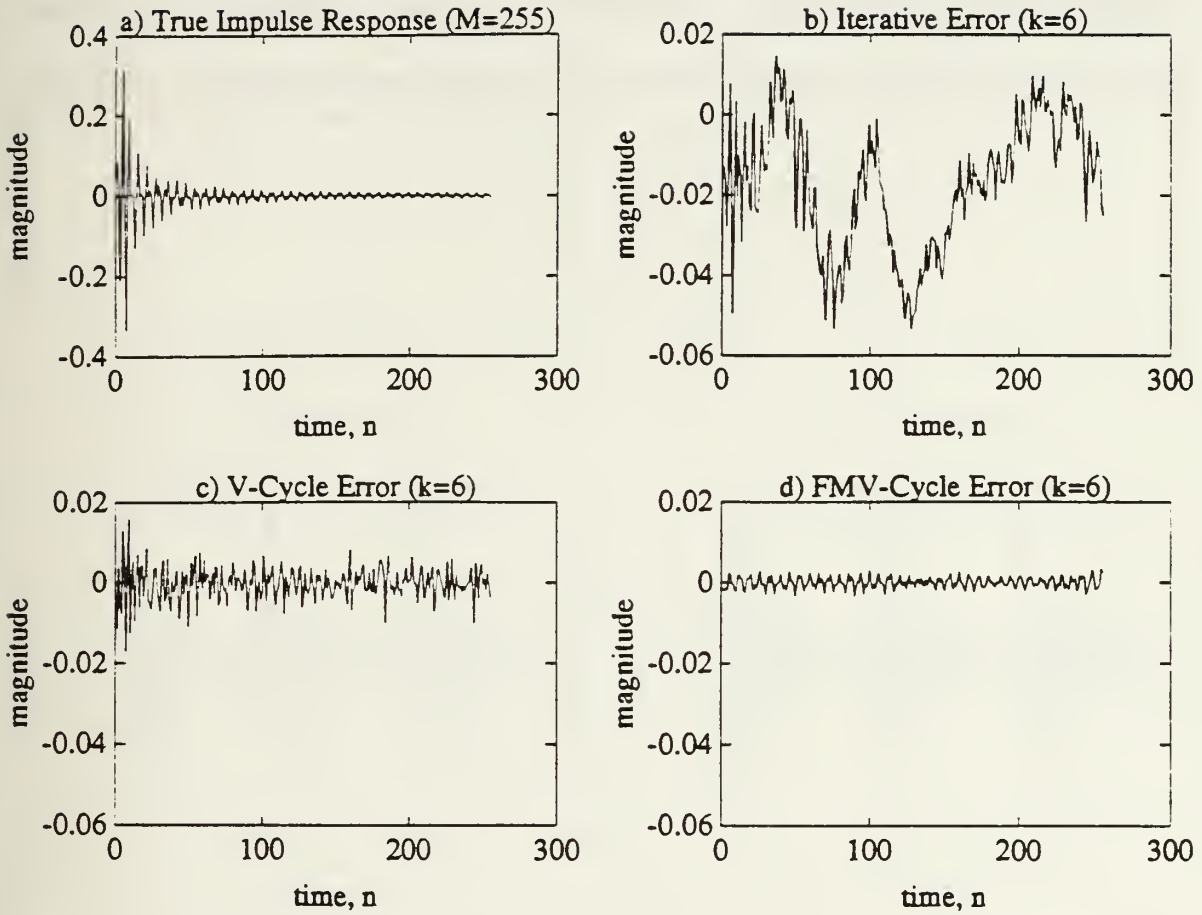

Figure A.12: 255th Order FIR Model Impulse Response Error Comparison. a) True System Impulse Response; b) Iterative Method Error; c) *V-cycle* Method Error; d) *FMV-cycle* Method Error.

APPENDIX B: AR SIMULATIONS

This appendix provides further AR modeling simulations of the voiced speech system described in Chapter IV. Filter orders of 3, 7, 31, and 63 are presented using a frame of voiced speech data of $L=300$ points as the system. Graphical results of these filters are provided for the straight iterative, the *V-cycle*, and the *FMV-cycle* methods.

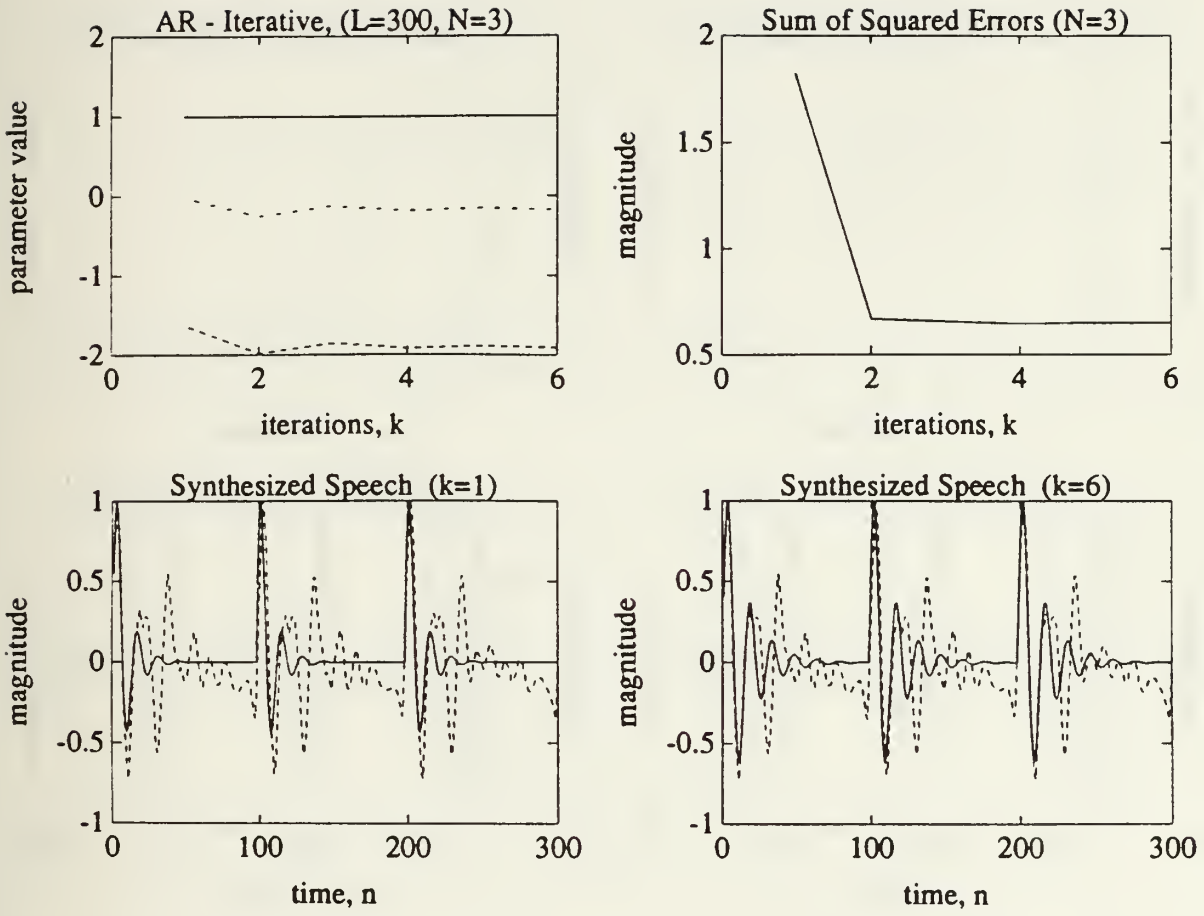

Figure B.1: 3rd Order AR Model of Voiced Speech Using the Straight Iterative Method. Original speech waveform shown in dashed lines, synthesized speech waveform shown in solid lines.

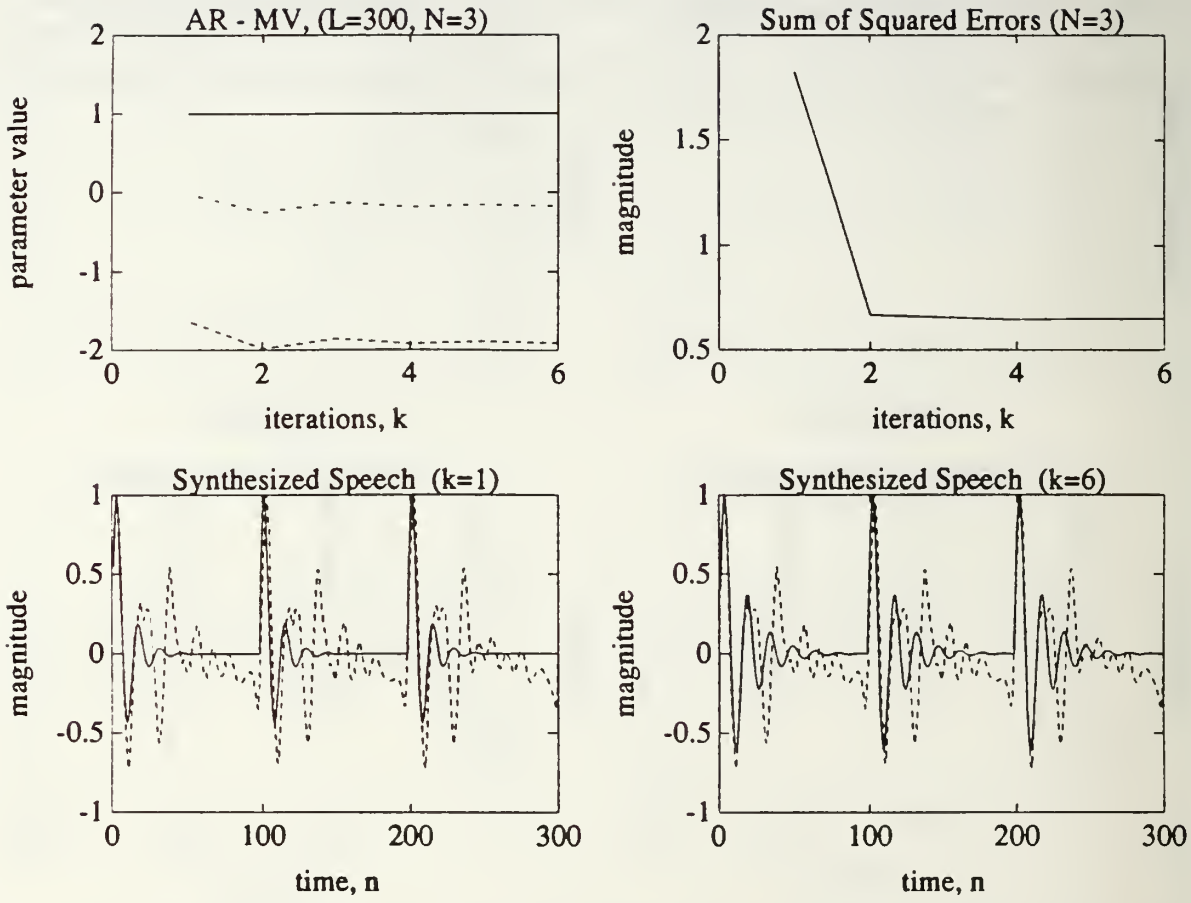

Figure B.2: 3rd Order AR Model of Voiced Speech Using the *V-cycle* Method. Original speech waveform shown in dashed lines, synthesized speech waveform shown in solid lines.

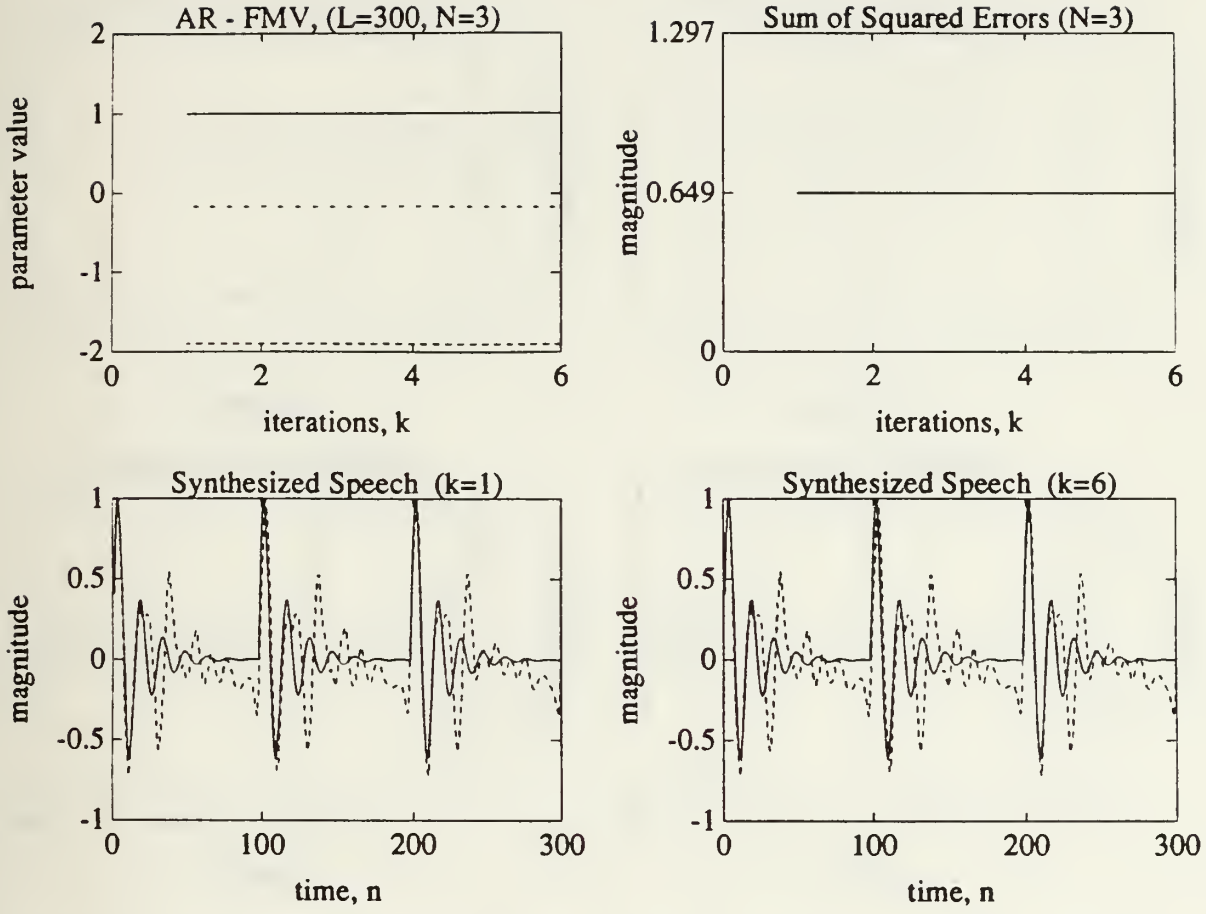

Figure B.3: 3rd Order AR Model of Voiced Speech Using the *FMV-cycle* Method. Original speech waveform shown in dashed lines, synthesized waveform shown in solid lines.

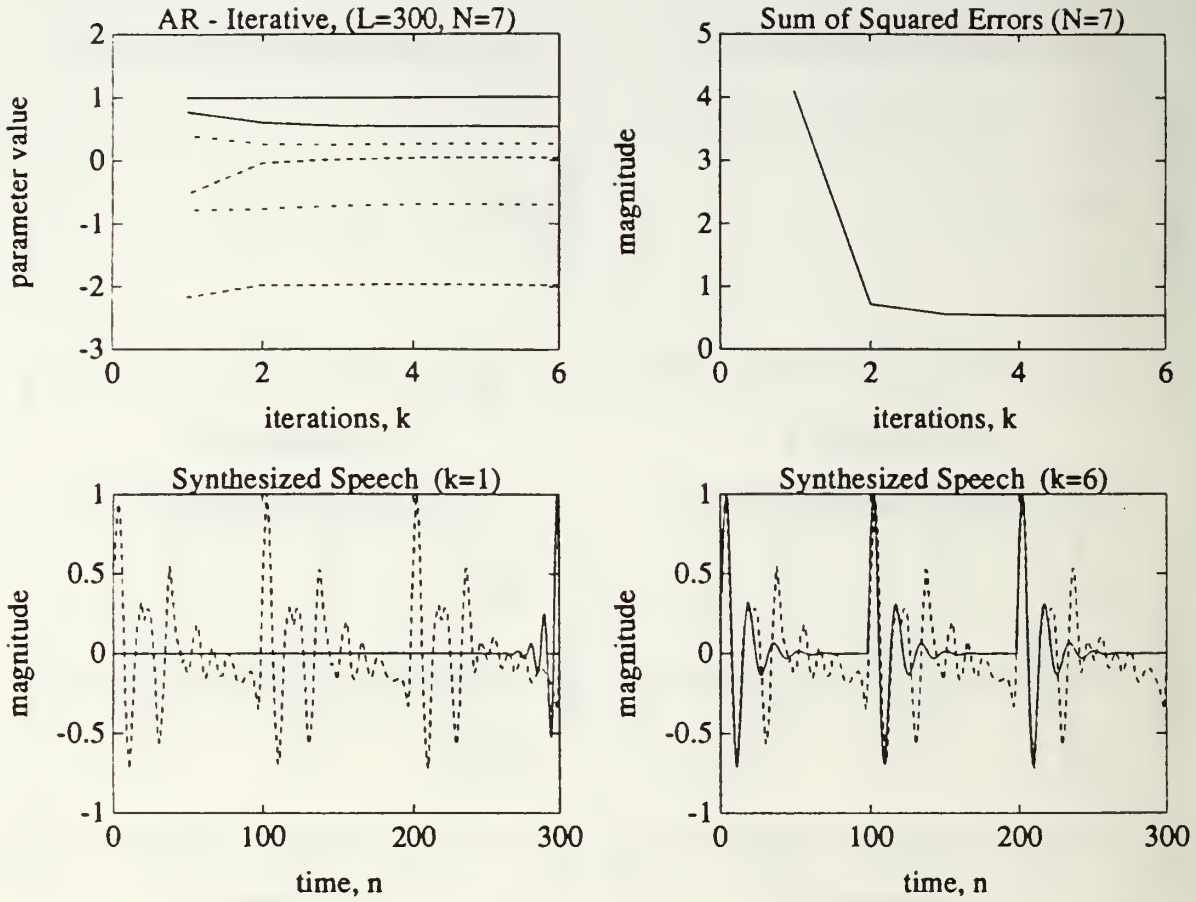

Figure B.4: 7th Order AR Model of Voiced Speech Using the Straight Iterative Method. Original speech waveform shown in dashed lines, synthesized speech waveform shown in solid lines.

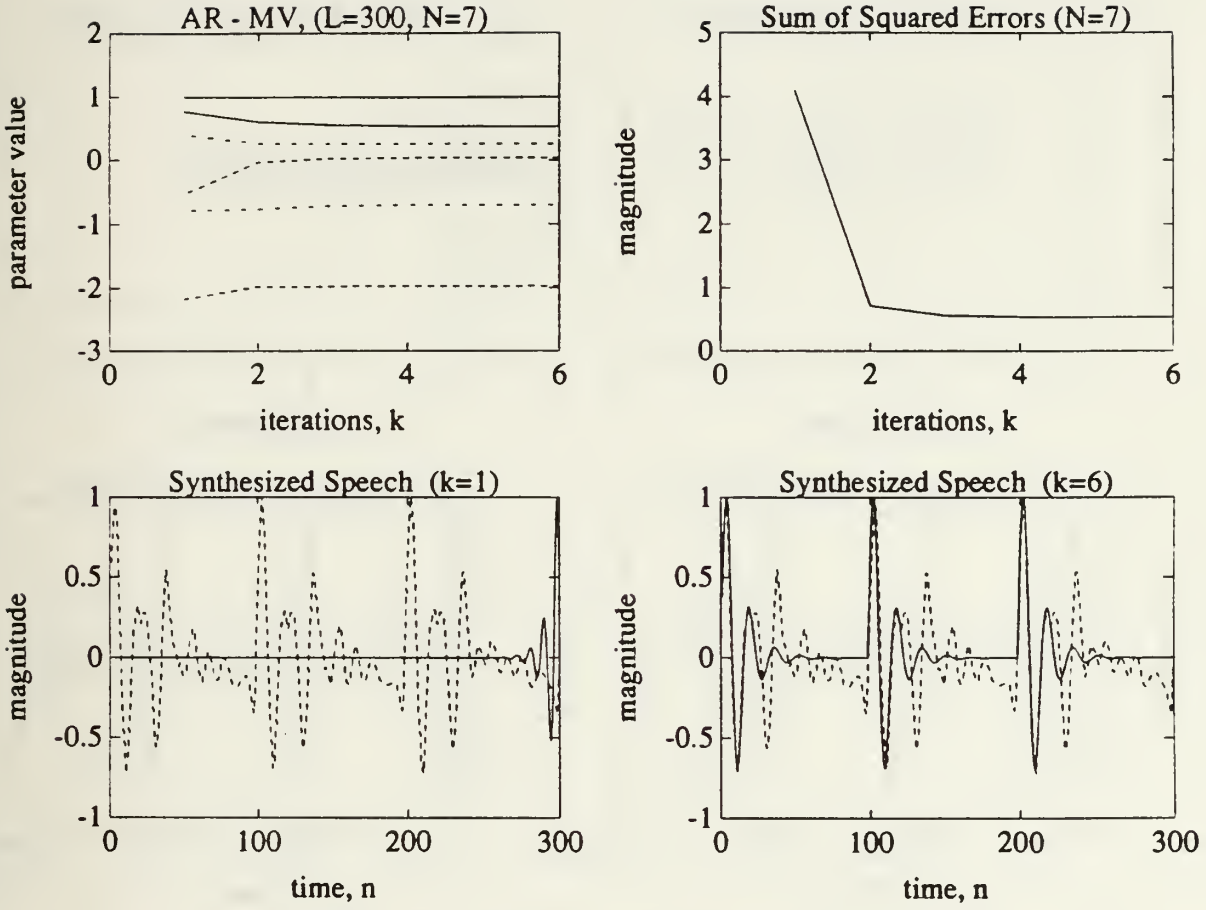

Figure B.5: 7th Order AR Model of Voiced Speech Using the *V-cycle* Method. Original speech waveform shown in dashed lines, synthesized speech waveform shown in solid lines.

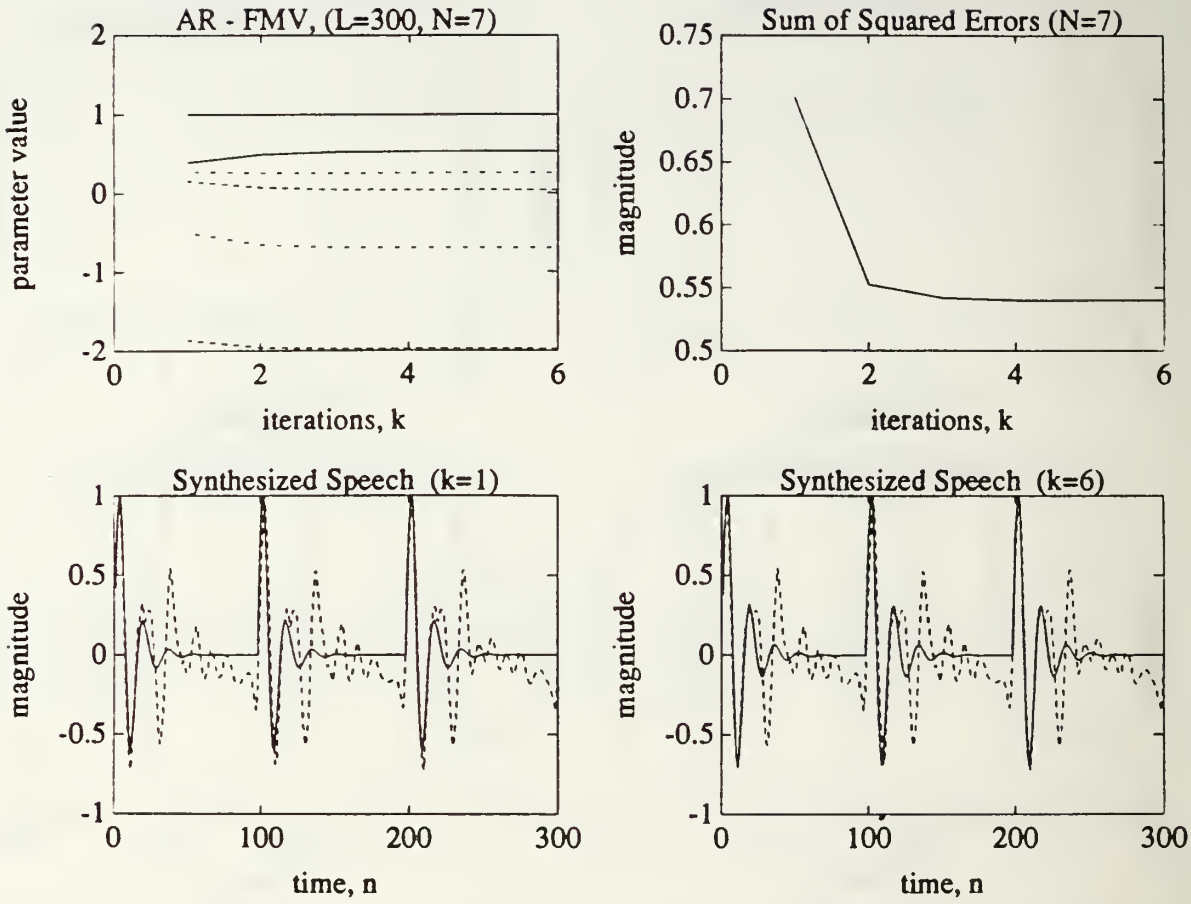

Figure B.6: 7th Order AR Model of Voiced Speech Using the *FMV-cycle* Method. Original speech waveform shown in dashed lines, synthesized speech waveform shown in solid lines.

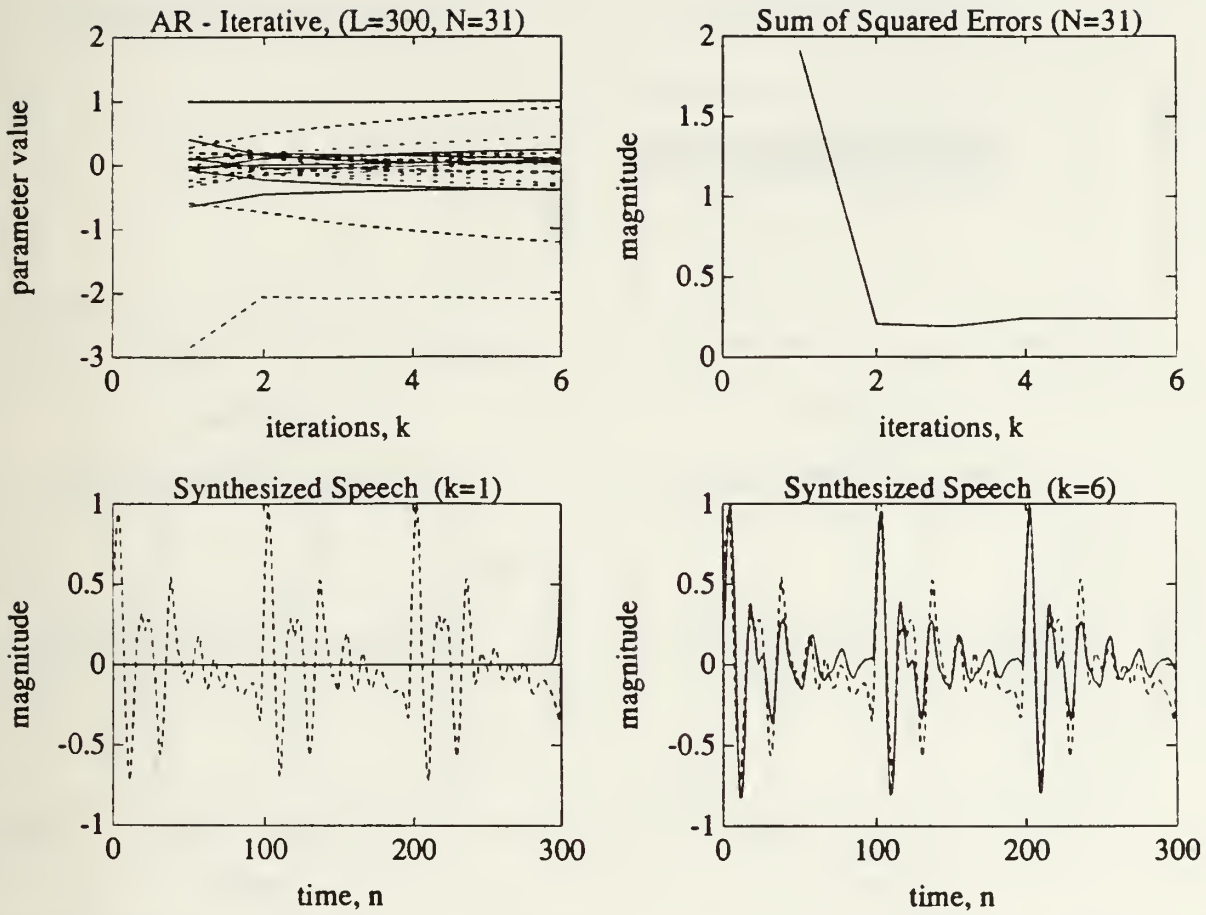

Figure B.7: 31st Order AR Model of Voiced Speech Using the Straight Iterative Method. Original speech waveform shown in dashed lines, synthesized speech waveform shown in solid lines.

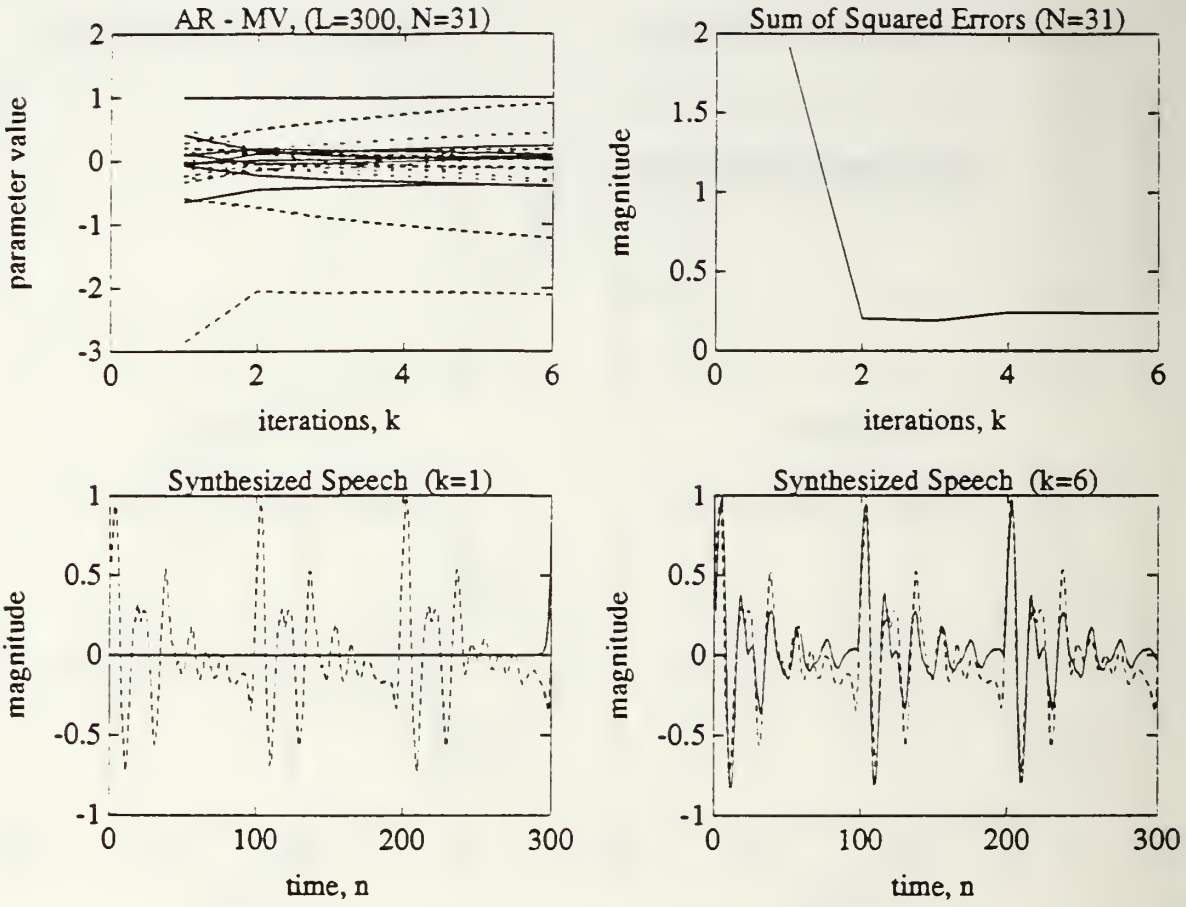

Figure B.8: 31st Order AR Model of Voiced Speech Using the *V-cycle* Method. Original speech waveform shown in dashed lines, synthesized speech waveform shown in solid lines.

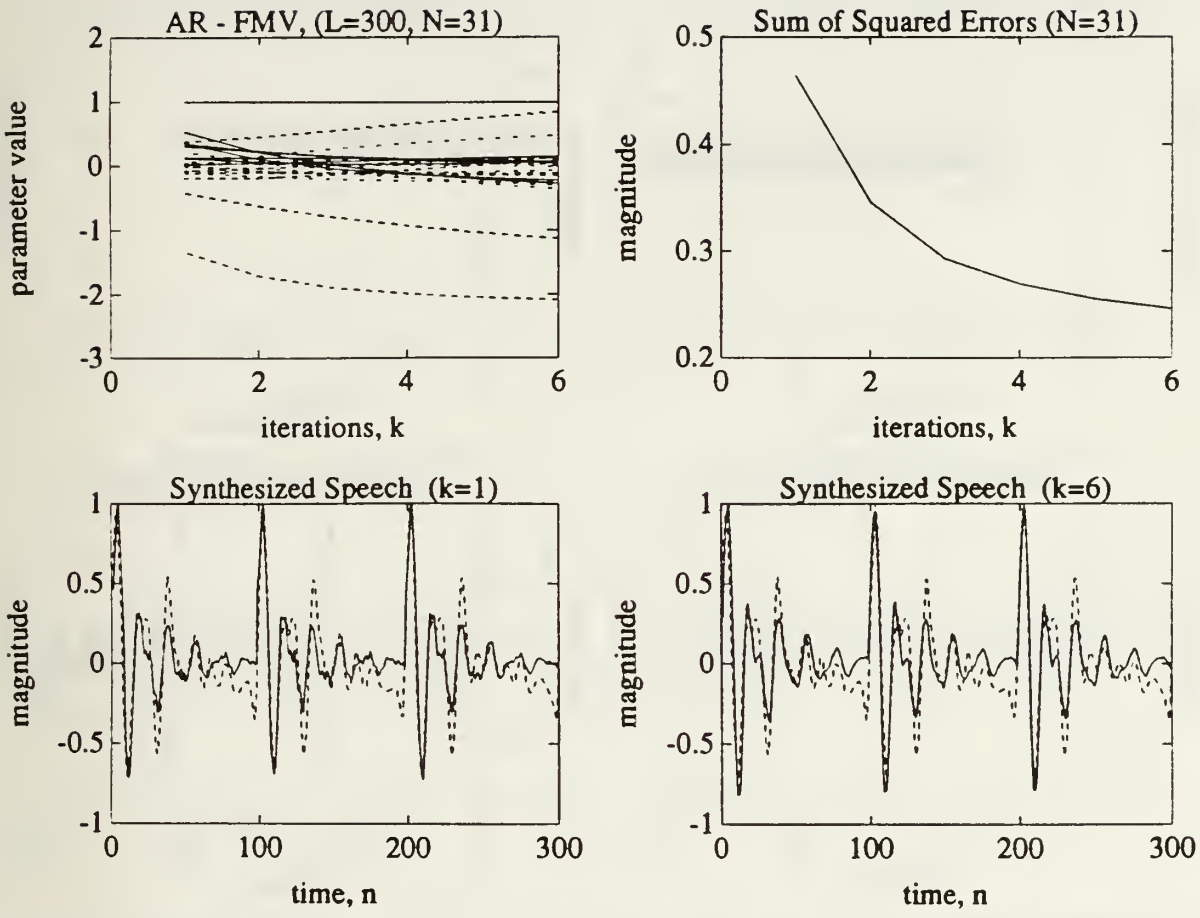

Figure B.9: 31st Order AR Model of Voiced Speech Using the *FMV-cycle* Method. Original speech waveform shown in dashed lines, synthesized speech waveform shown in solid lines.

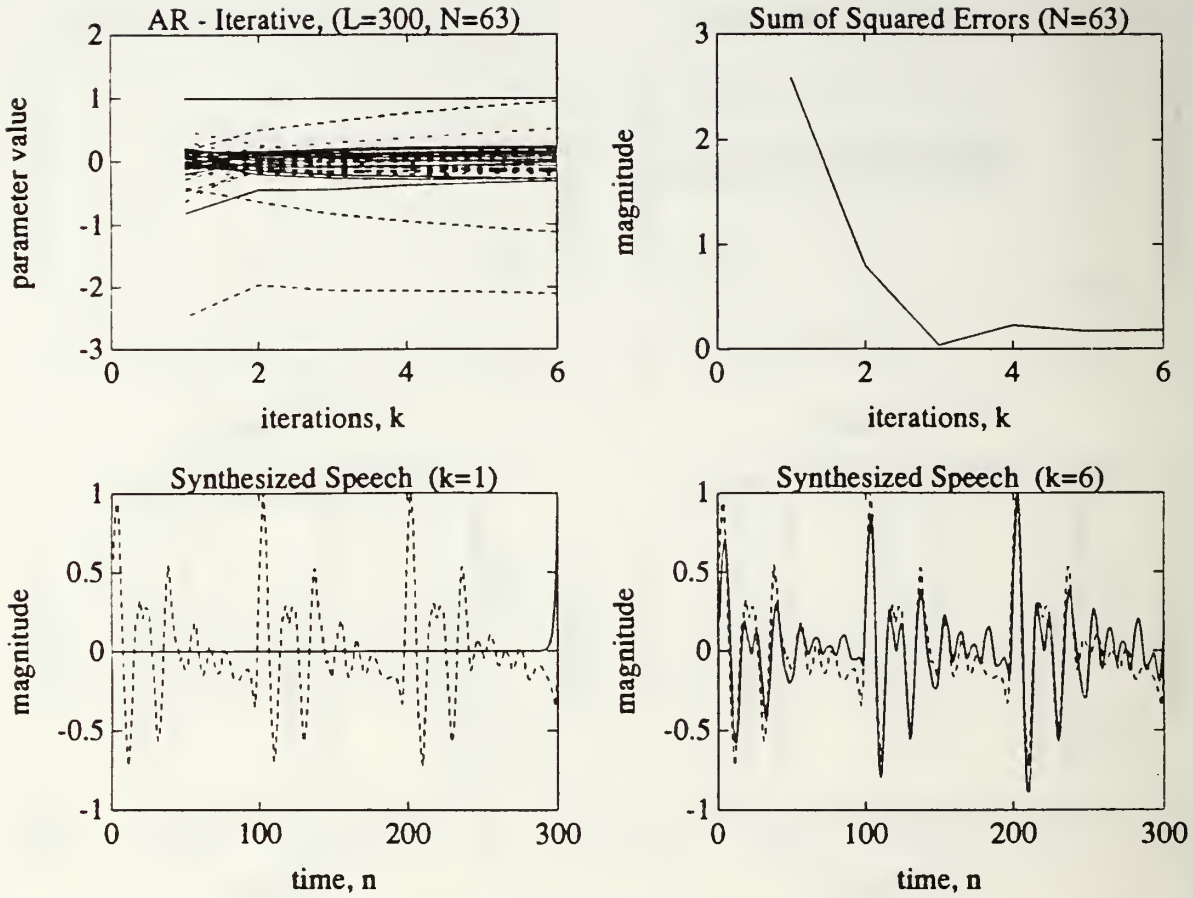

Figure B.10: 63rd Order AR Model of Voiced Speech Using the Straight Iterative Method. Original speech waveform shown in dashed lines, synthesized speech waveform shown in solid lines.

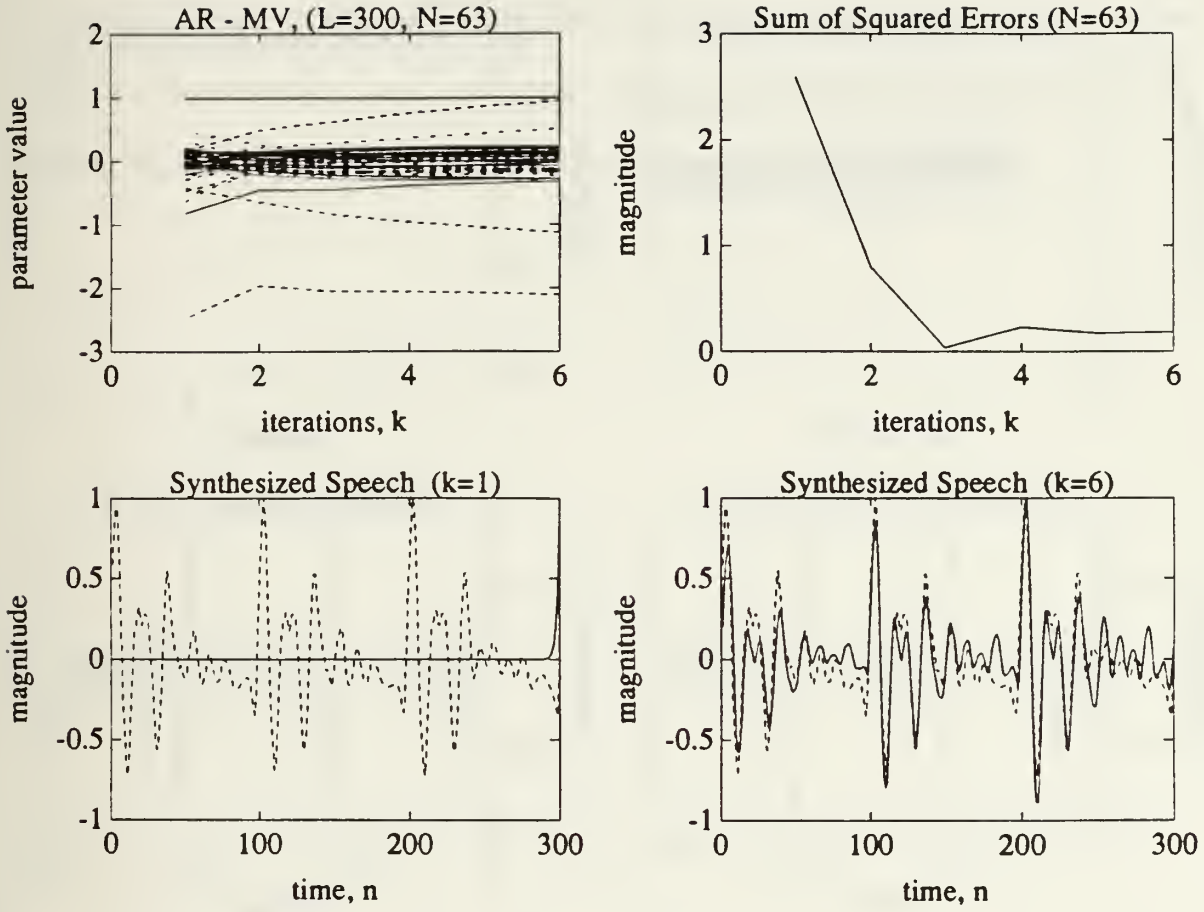

Figure B.11: 63rd Order AR Model of Voiced Speech Using the *V-cycle* Method. Original speech waveform shown in dashed lines, synthesized speech waveform shown in solid lines.

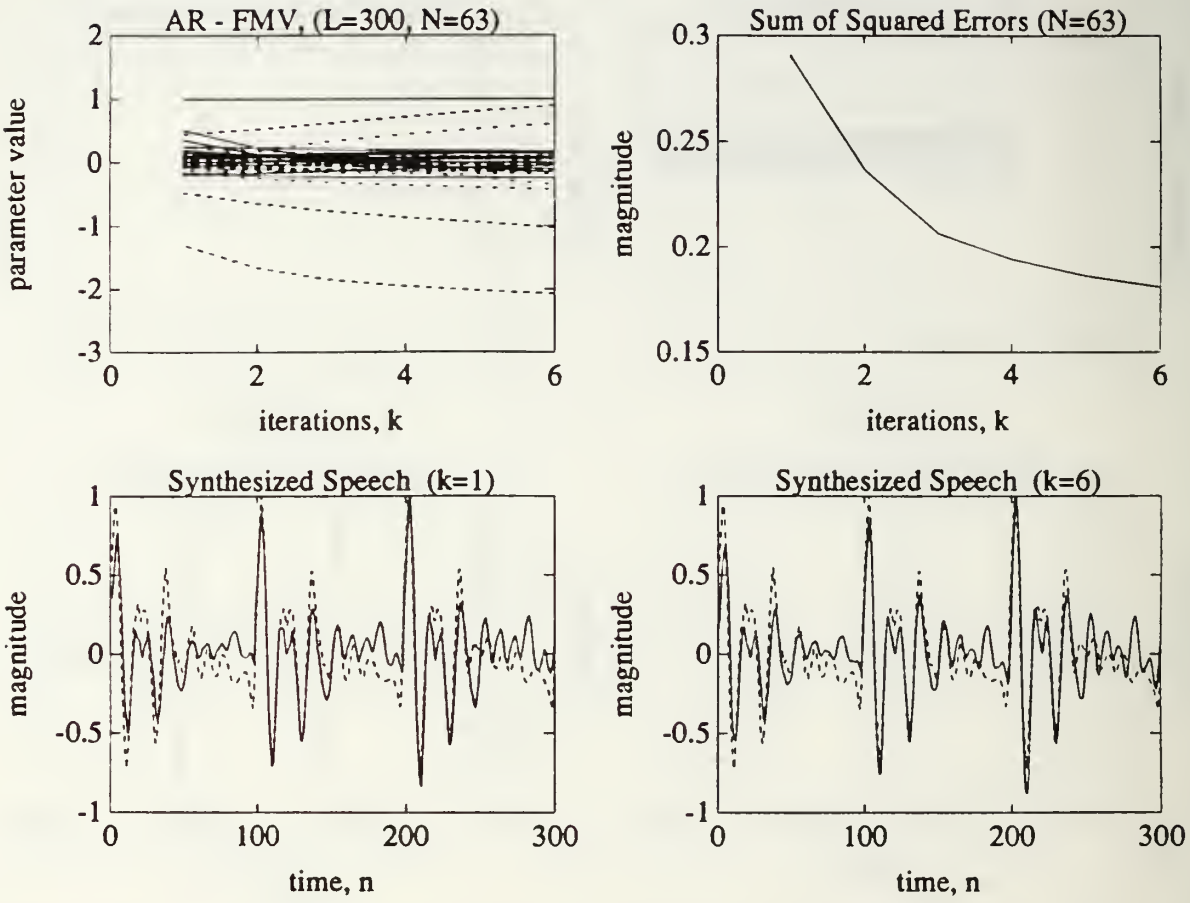

Figure B.12: 63rd Order AR Model of Voiced Speech Using the *FMV-cycle* Method. Original speech waveform shown in dashed lines, synthesized speech waveform shown in solid lines.

APPENDIX C: IIR SIMULATIONS

This appendix provides further IIR modeling simulations of the system described in Chapter IV. Filter orders of 7, 15, 23, and 63 are presented using a data sequence of $L=1000$ points created by filtering zero mean, unit variance white noise with the bandpass filter of Figure 4.1. Graphical results of these filters are provided for the straight iterative, the *V-cycle*, and the *FMV-cycle* methods.

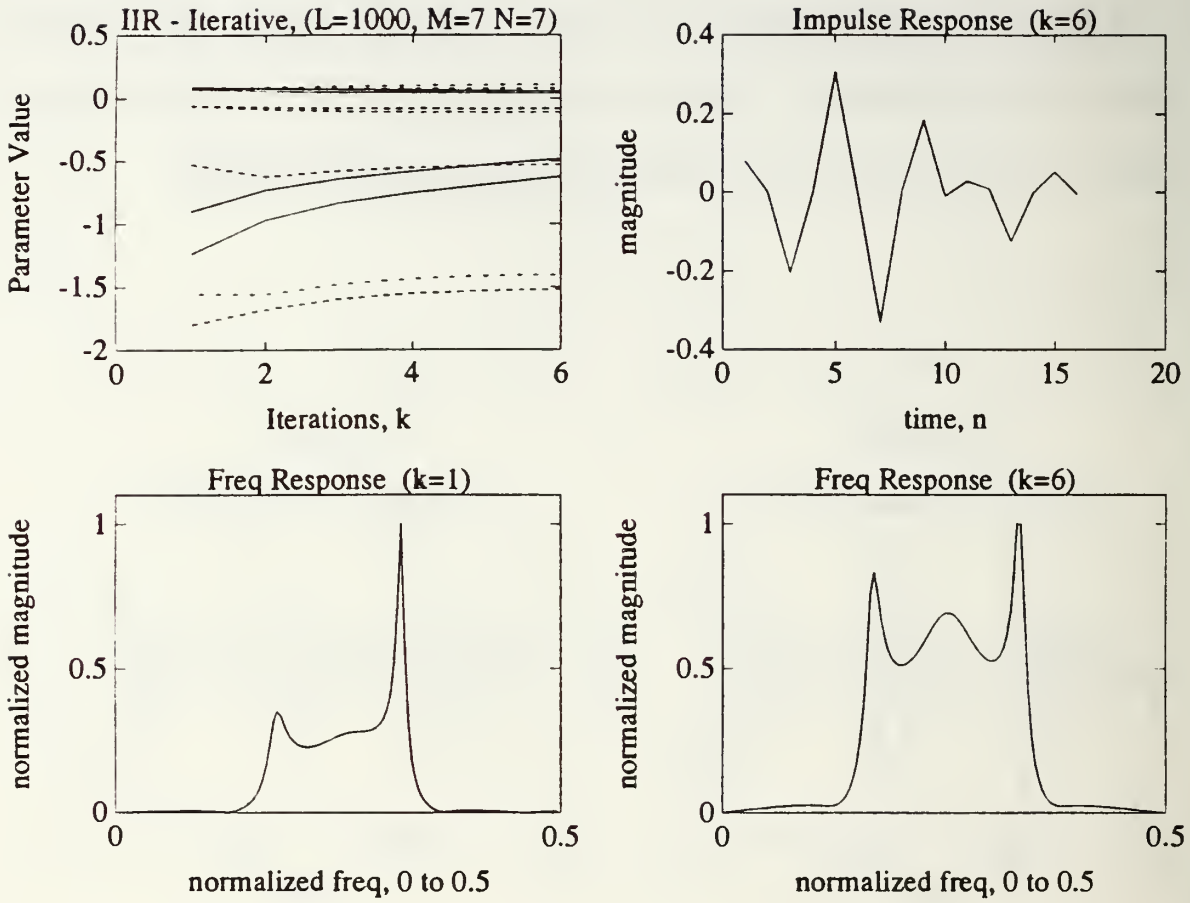

Figure C.1: 7th Order IIR Model Using the Straight Iterative Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

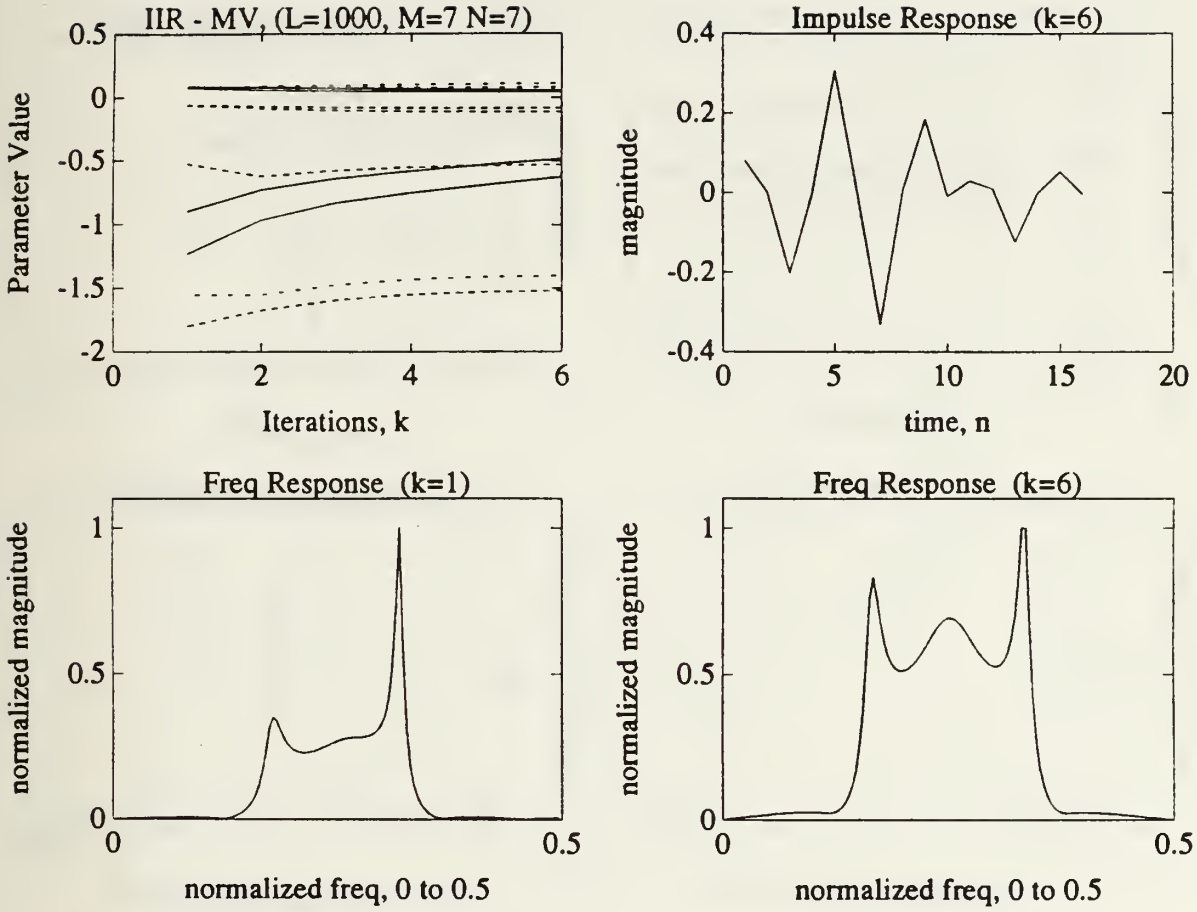

Figure C.2: 7th Order IIR Model Using the *V-cycle* Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

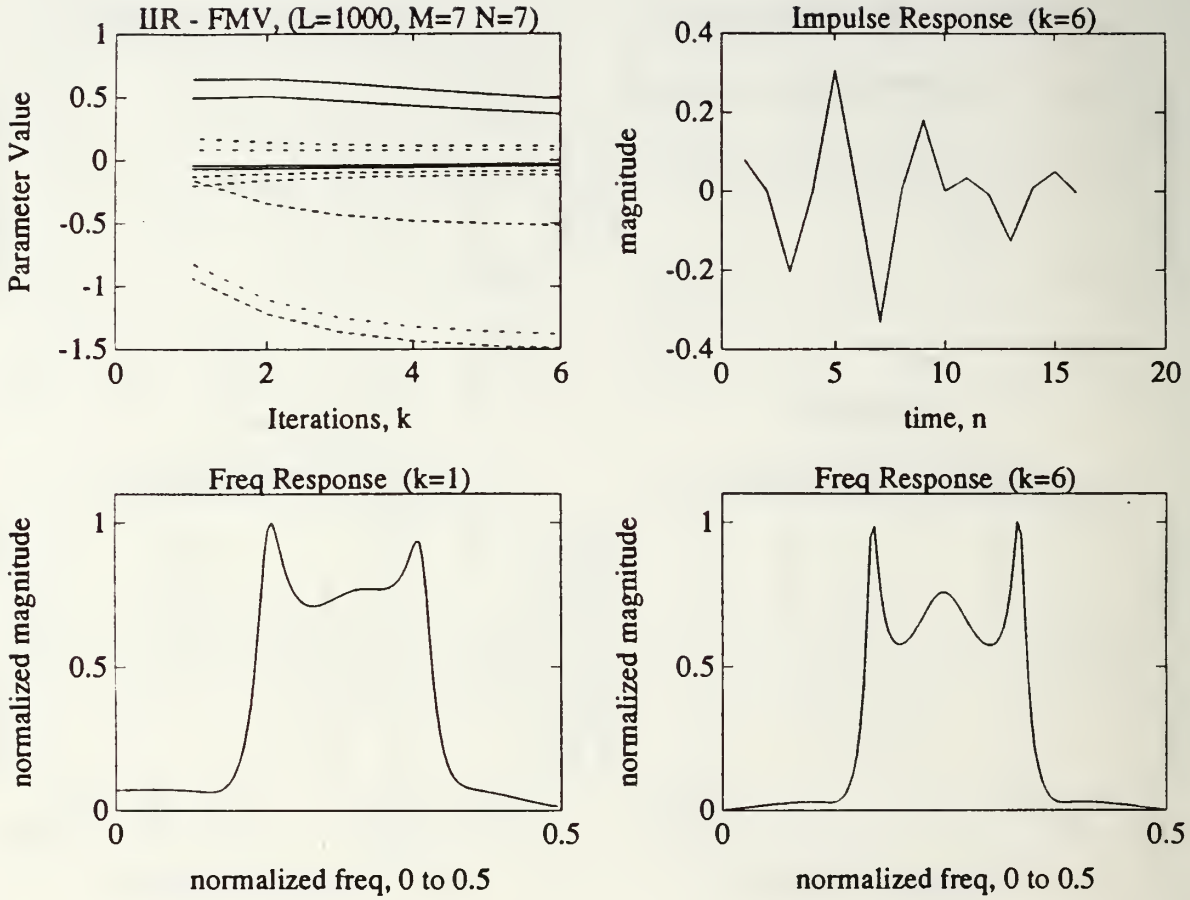

Figure C.3: 7th Order IIR Model Using the *FMV-cycle* Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

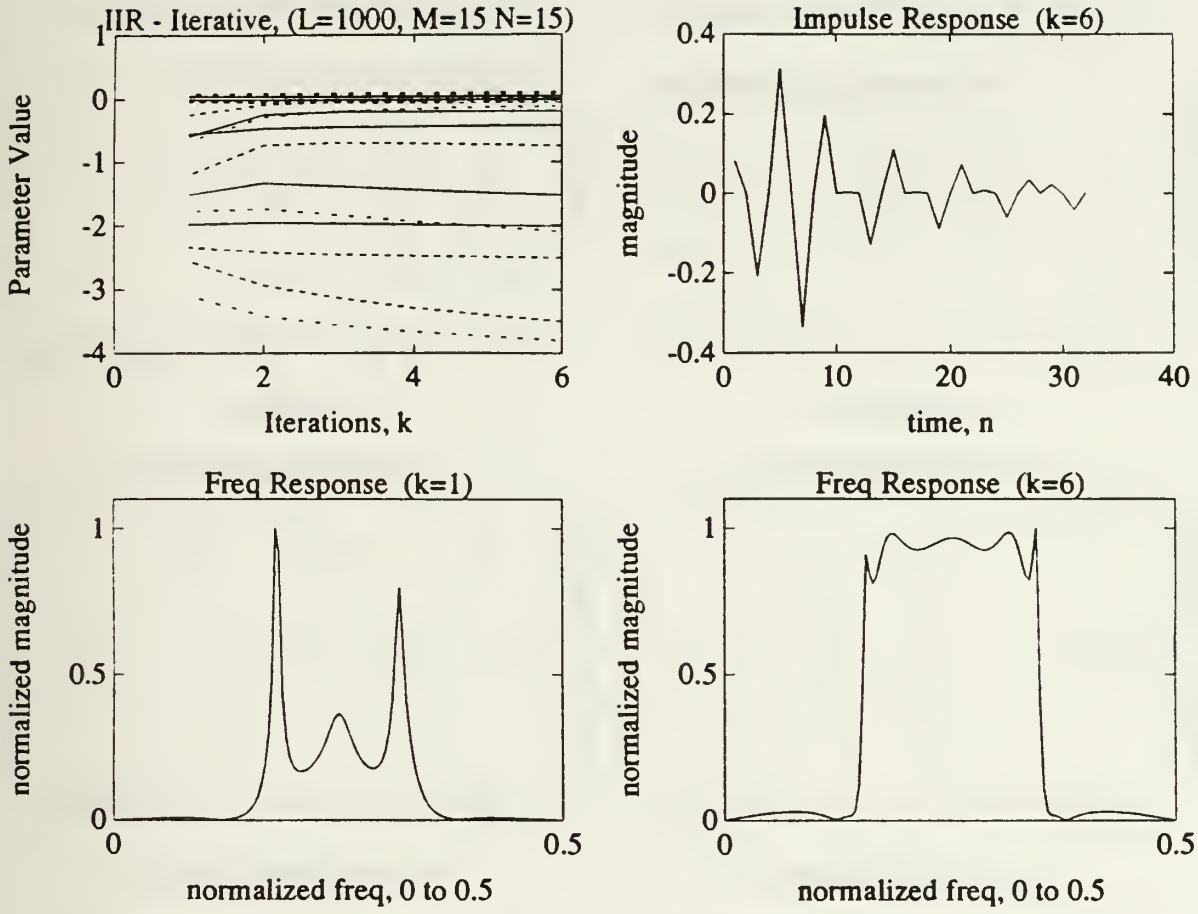

Figure C.4: 15th Order IIR Model Using the Straight Iterative Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

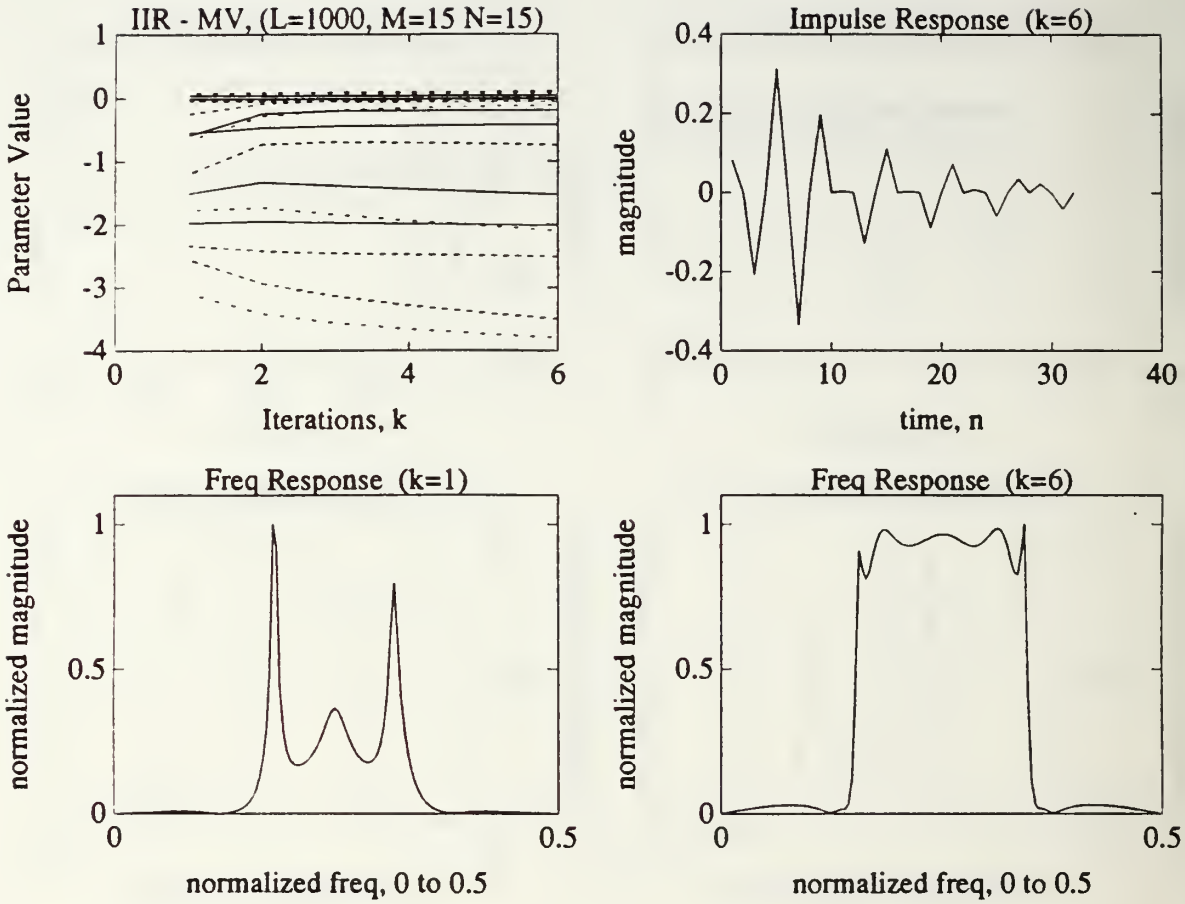

Figure C.5: 15th Order IIR Model Using the *V-cycle* Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

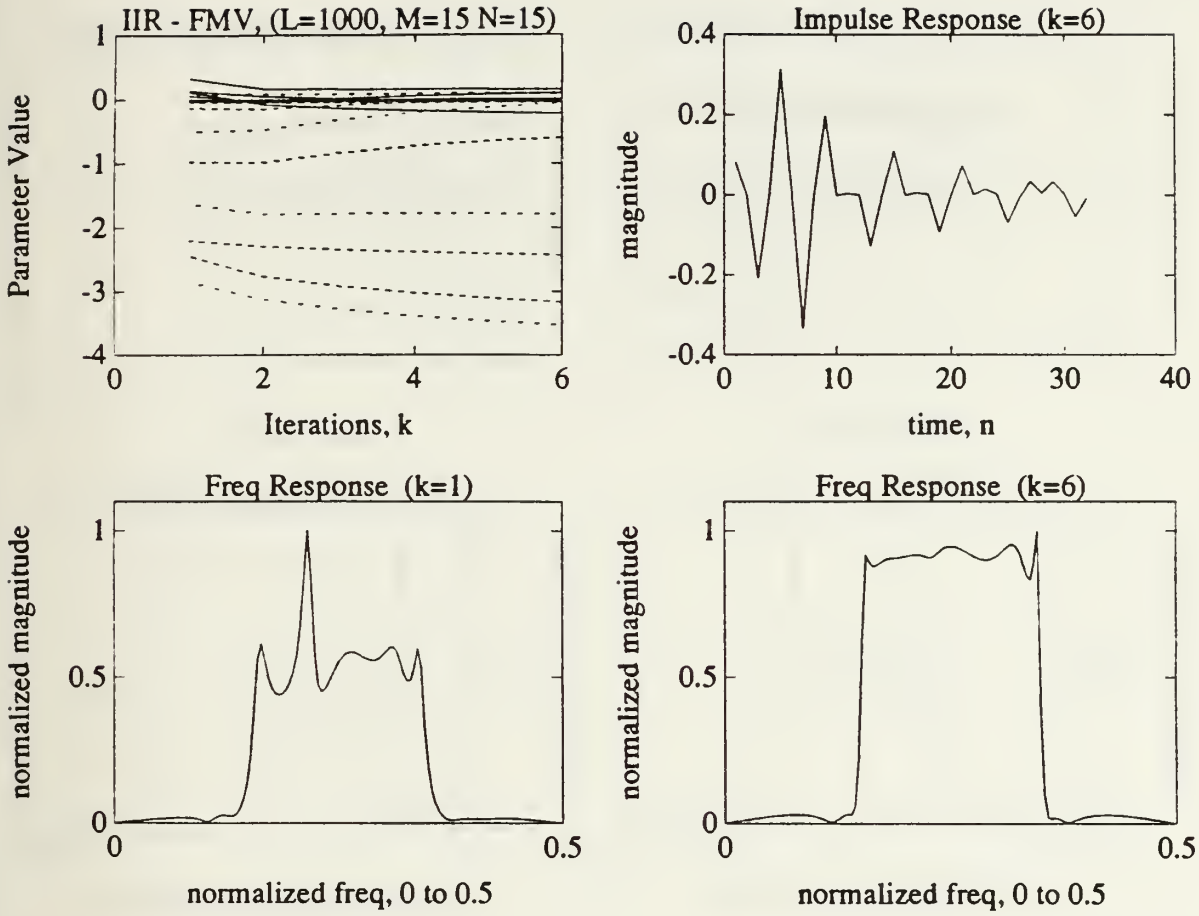

Figure C.6: 15th Order IIR Model Using the *FMV-cycle* Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

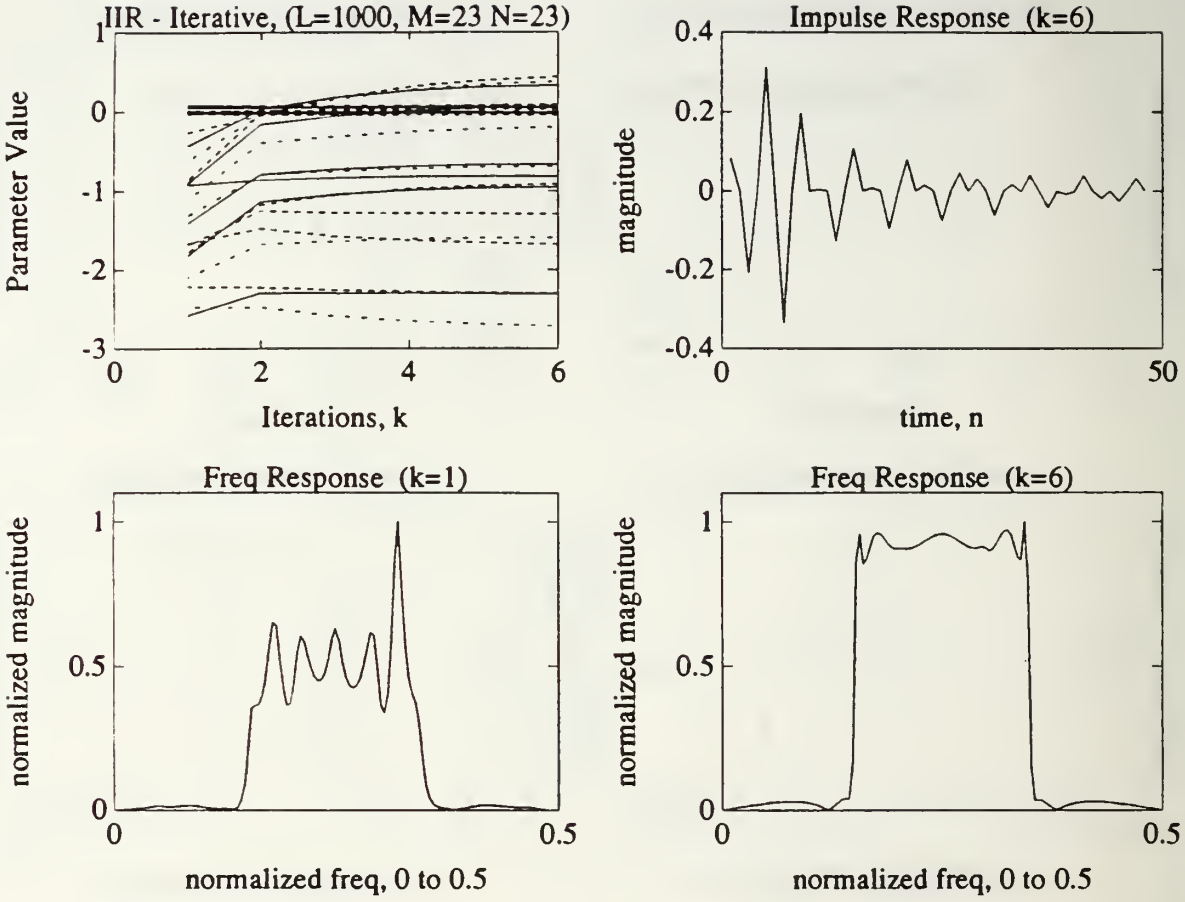

Figure C.7: 23rd Order IIR Model Using the Straight Iterative Method. Impulse and normalized frequency responses shown using $L = 1000$ data points.

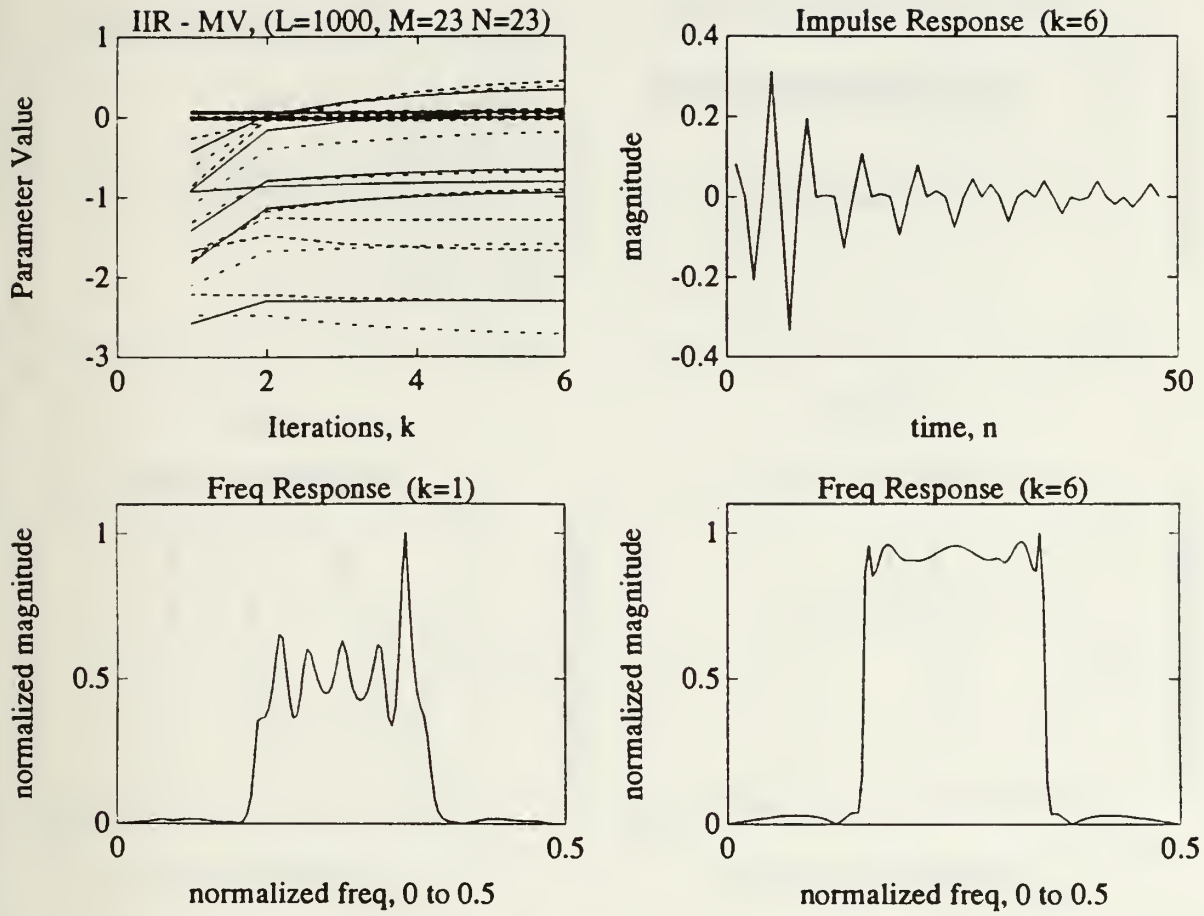

Figure C.8: 23rd Order IIR Model Using the *V-cycle* Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

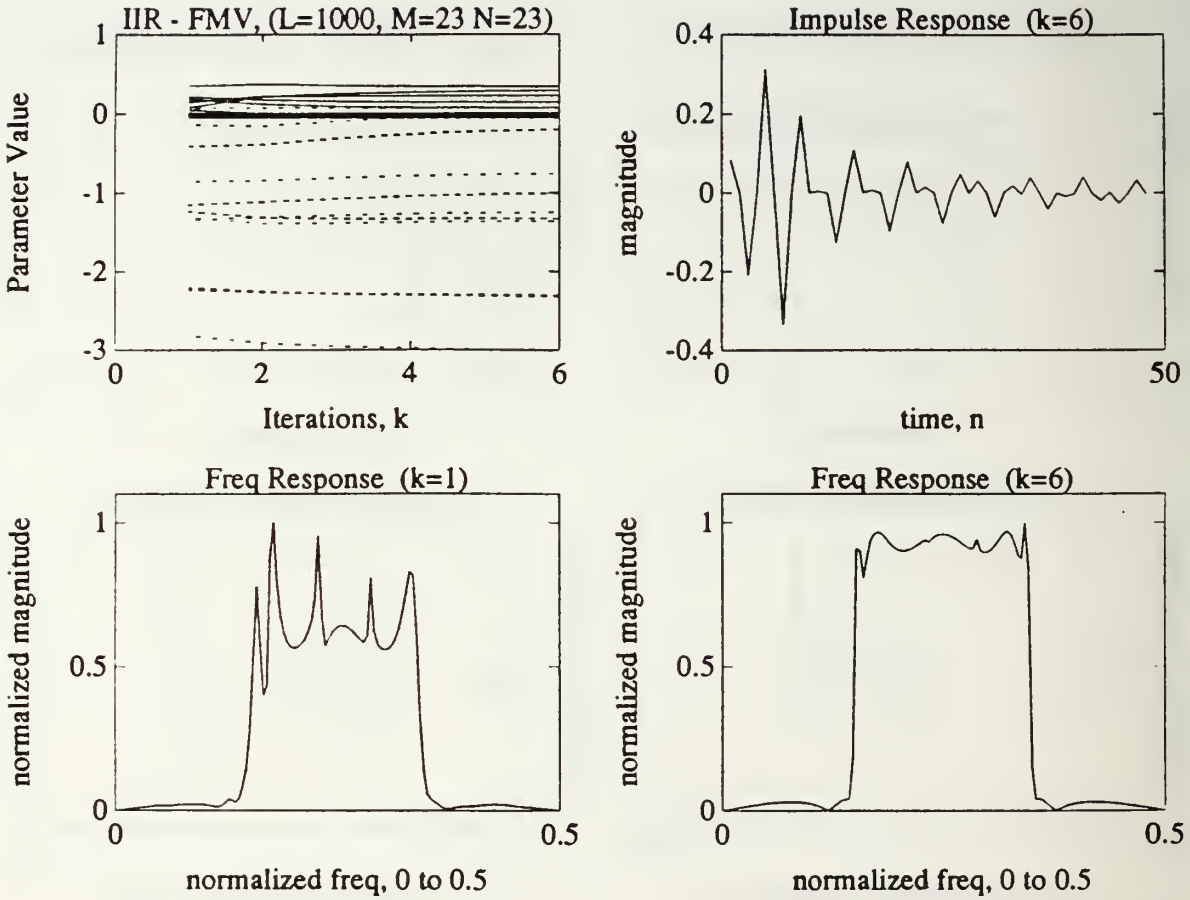

Figure C.9: 23rd Order IIR Model Using the *FMV-cycle* Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

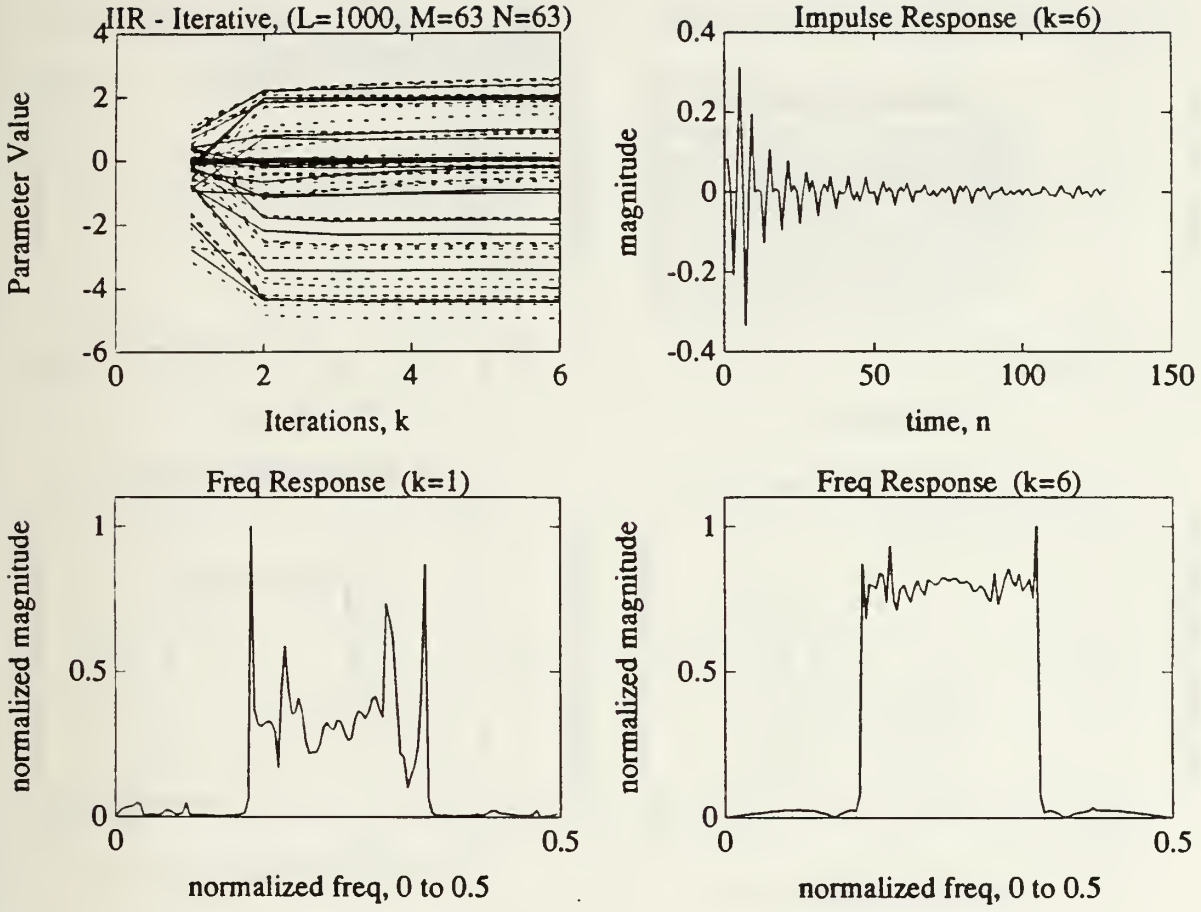

Figure C.10: 63rd Order IIR Model Using the Straight Iterative Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

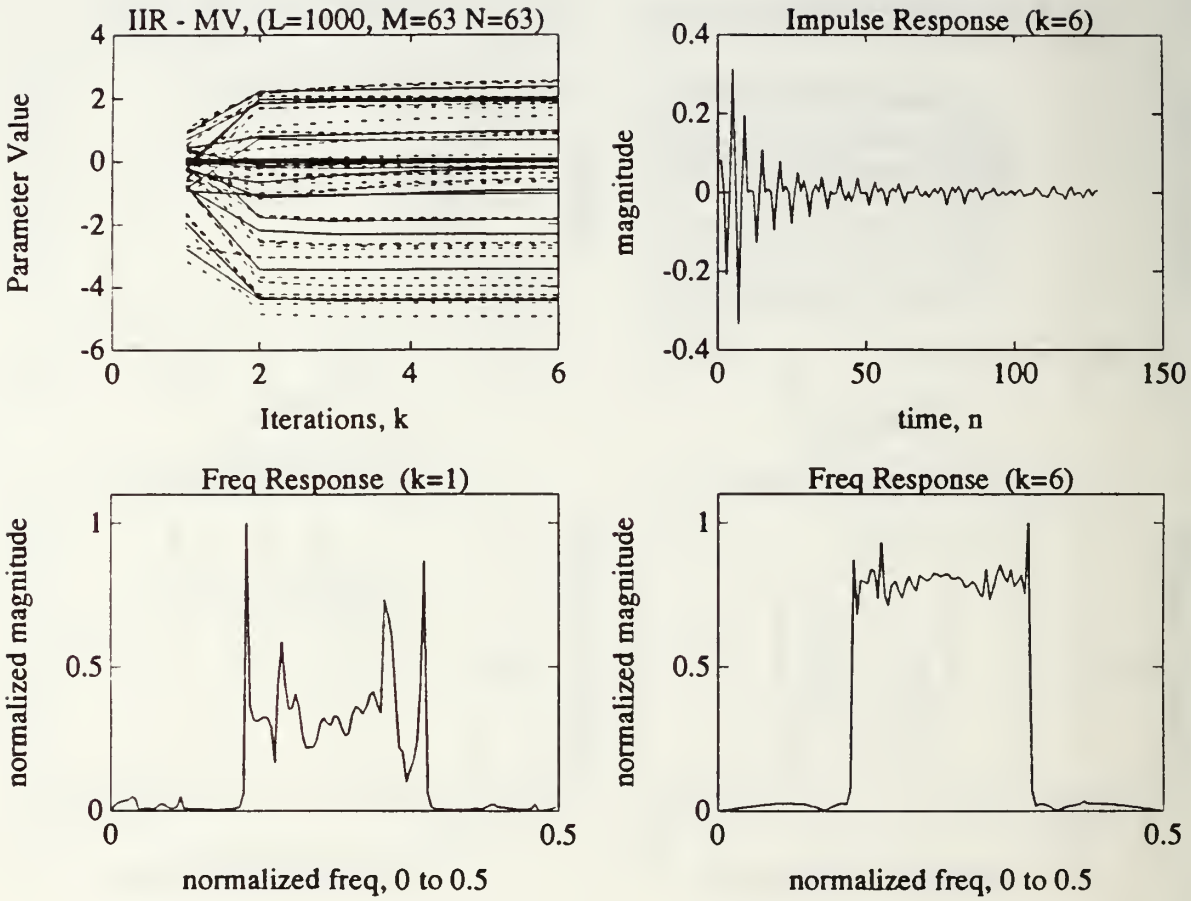

Figure C.11: 63rd Order IIR Model Using the *V-cycle* Method. Impulse and normalized frequency responses are shown using $L=1000$ data points.

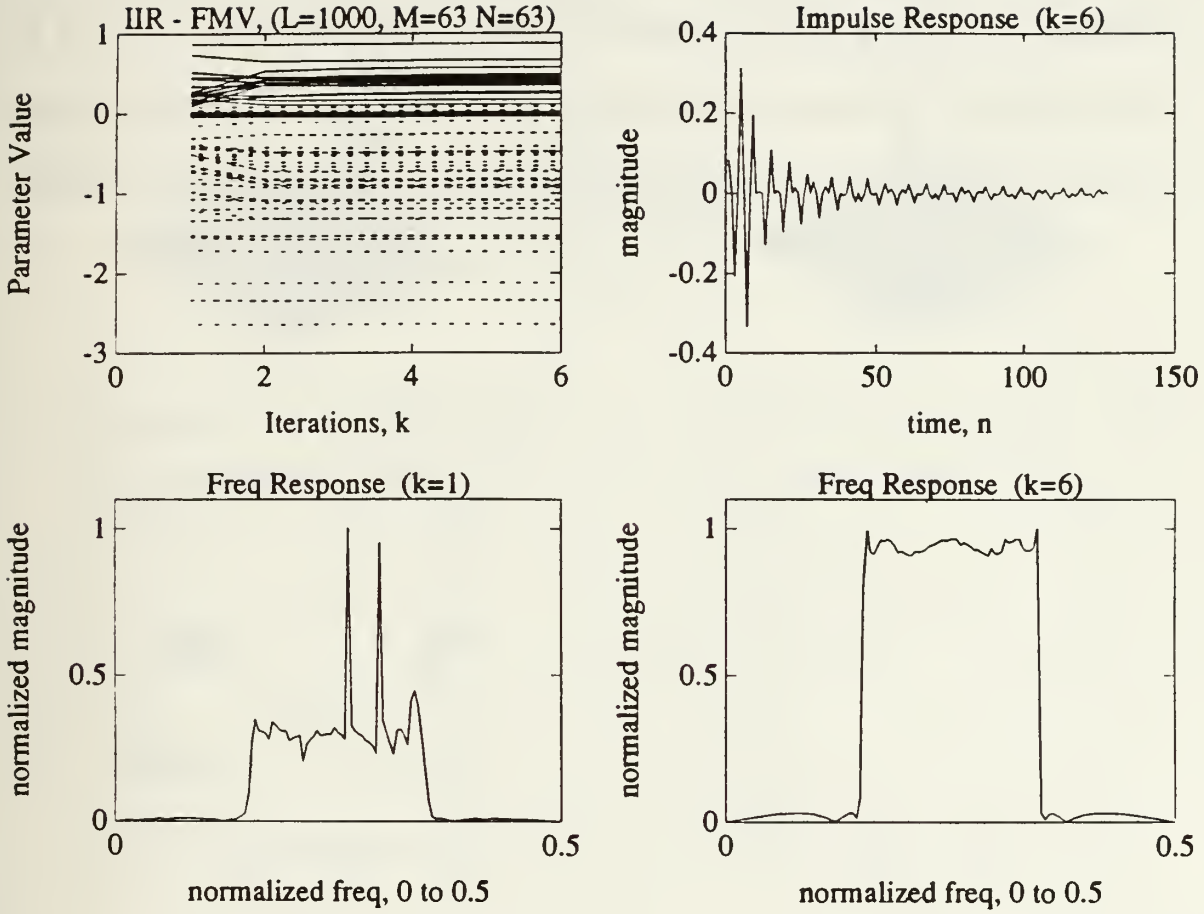

Figure C.12: 63rd Order IIR Model Using the *FMV-cycle* Method. Impulse and normalized frequency responses shown using $L=1000$ data points.

APPENDIX D: 2-D AR SPECTRAL ESTIMATION SIMULATIONS

This appendix provides further 2-D AR spectral estimation simulations of the system described in Chapter IV. The filter mask size of 9×9 is used to filter a 36×36 data sequence consisting of eight sinusoids in noise (SNR = 10dB) for $k = 1, 3, \& 10$ fine grid iterations. Graphical results of the 1st quadrant, 2nd quadrant, and CQ spectral estimates are provided for the straight iterative, the *V-cycle*, and the *FMV-cycle* methods.

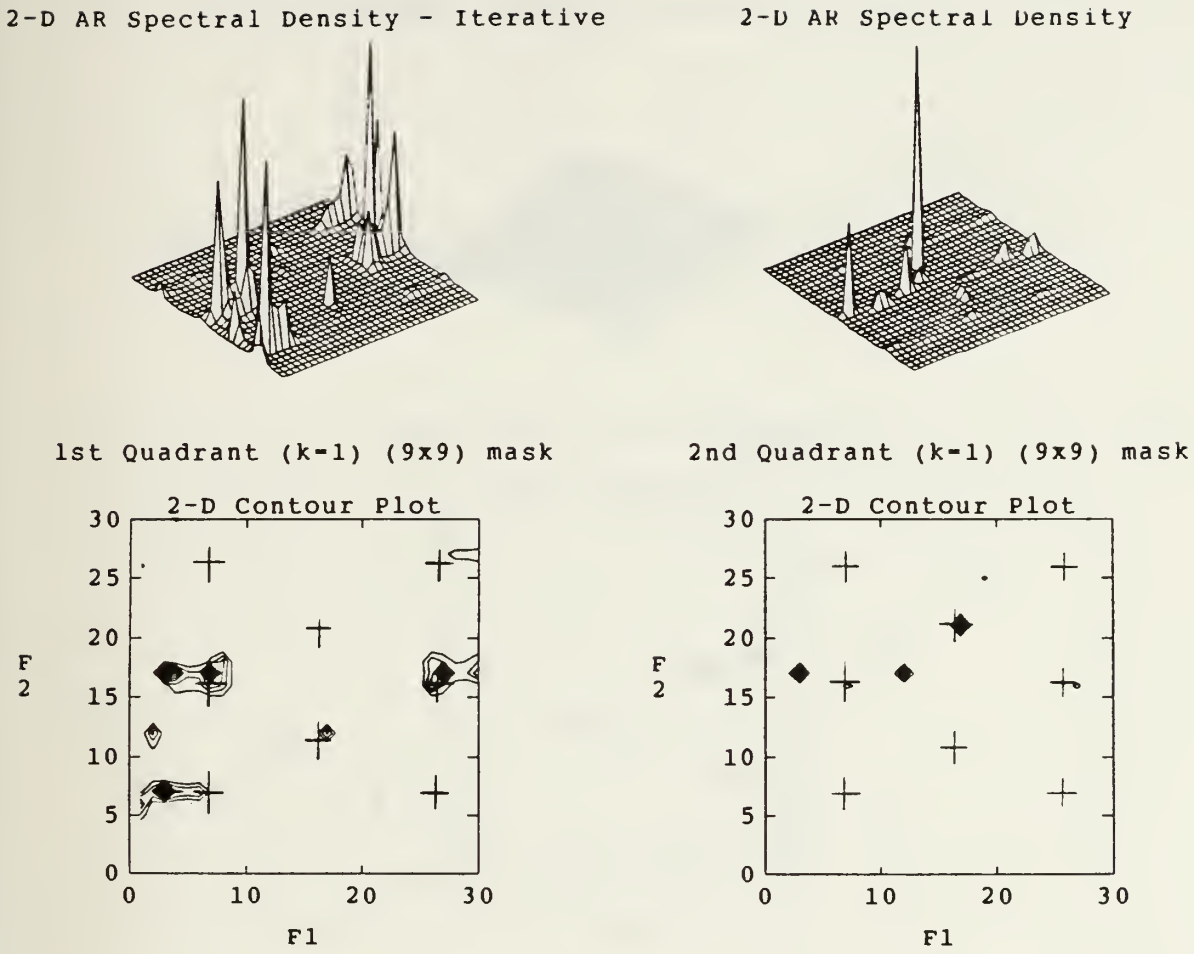

Figure D.1: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR=10dB) Using the Straight Iterative Method. 1st & 2nd quadrant estimates shown, (k=1, 9×9 mask).

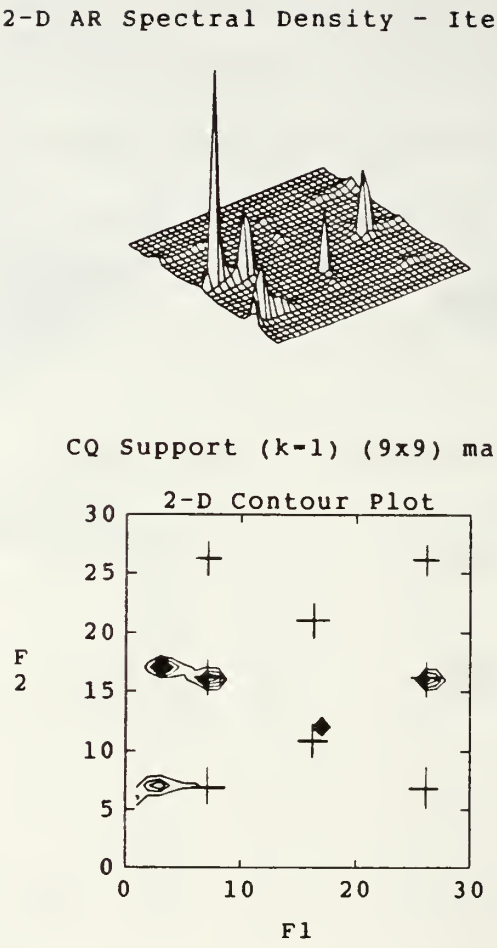

Figure D.2: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the Straight Iterative Method. CQ estimate shown, (k=1, 9×9 mask).

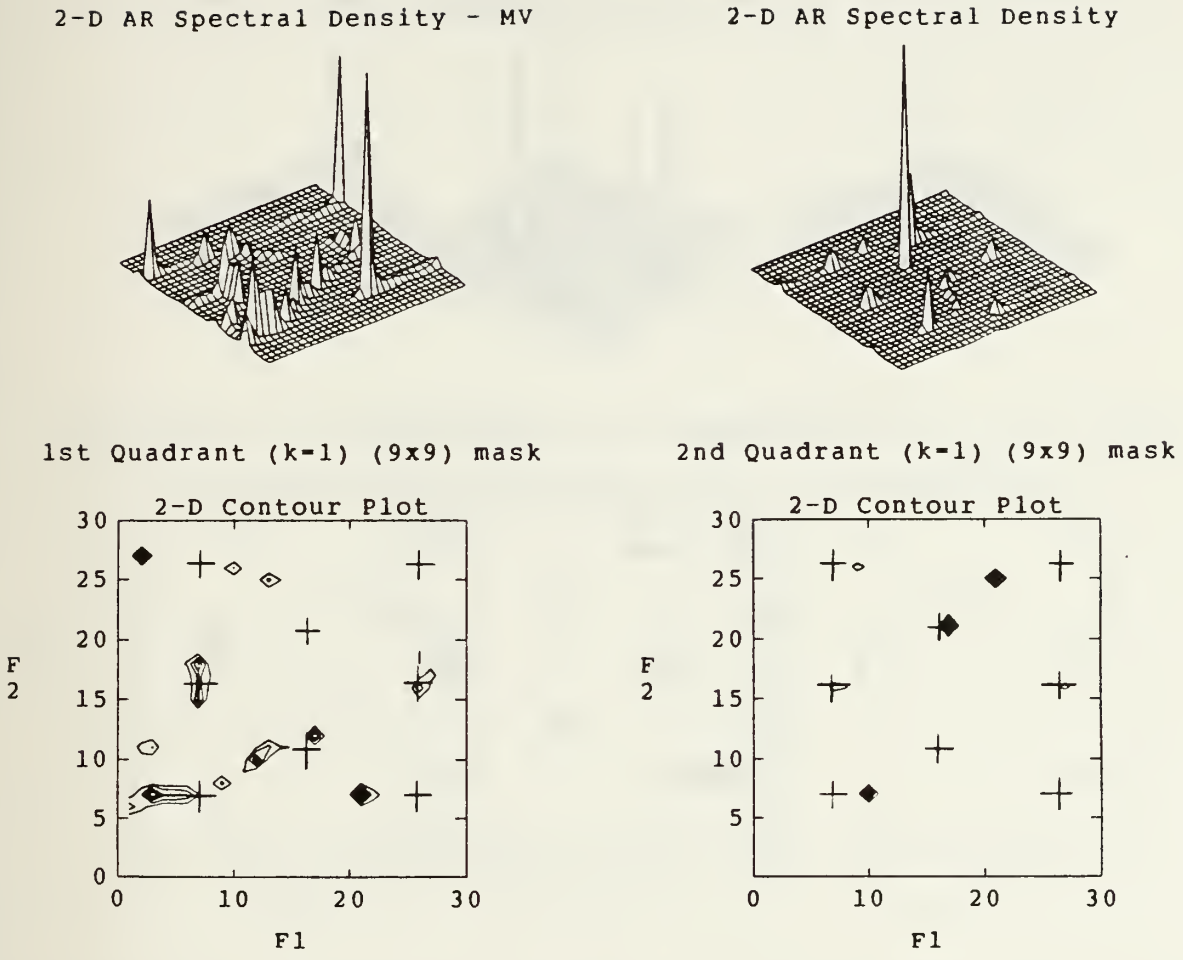

Figure D.3: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR=10dB) Using the *V-cycle* Method. 1st & 2nd quadrant estimates shown, (k=1, 9×9 mask).

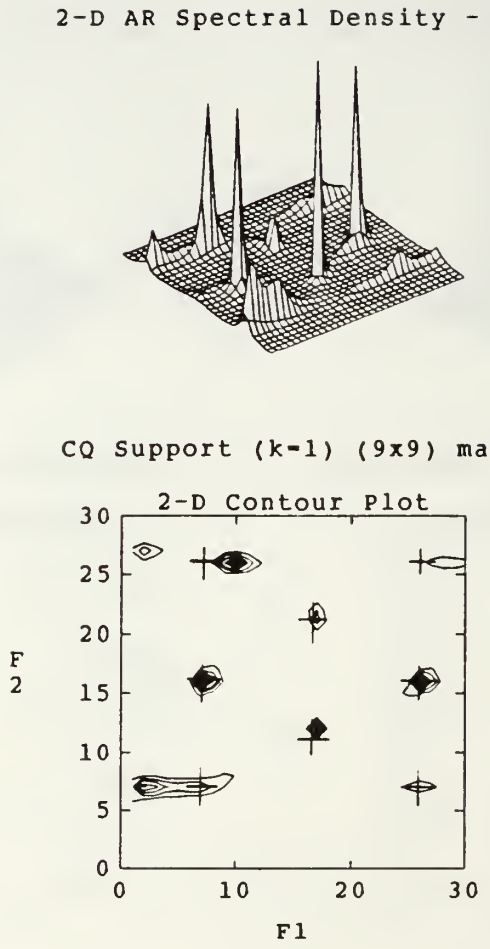

Figure D.4: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the *V-cycle*. CQ estimate shown, (k=1, 9×9 mask).

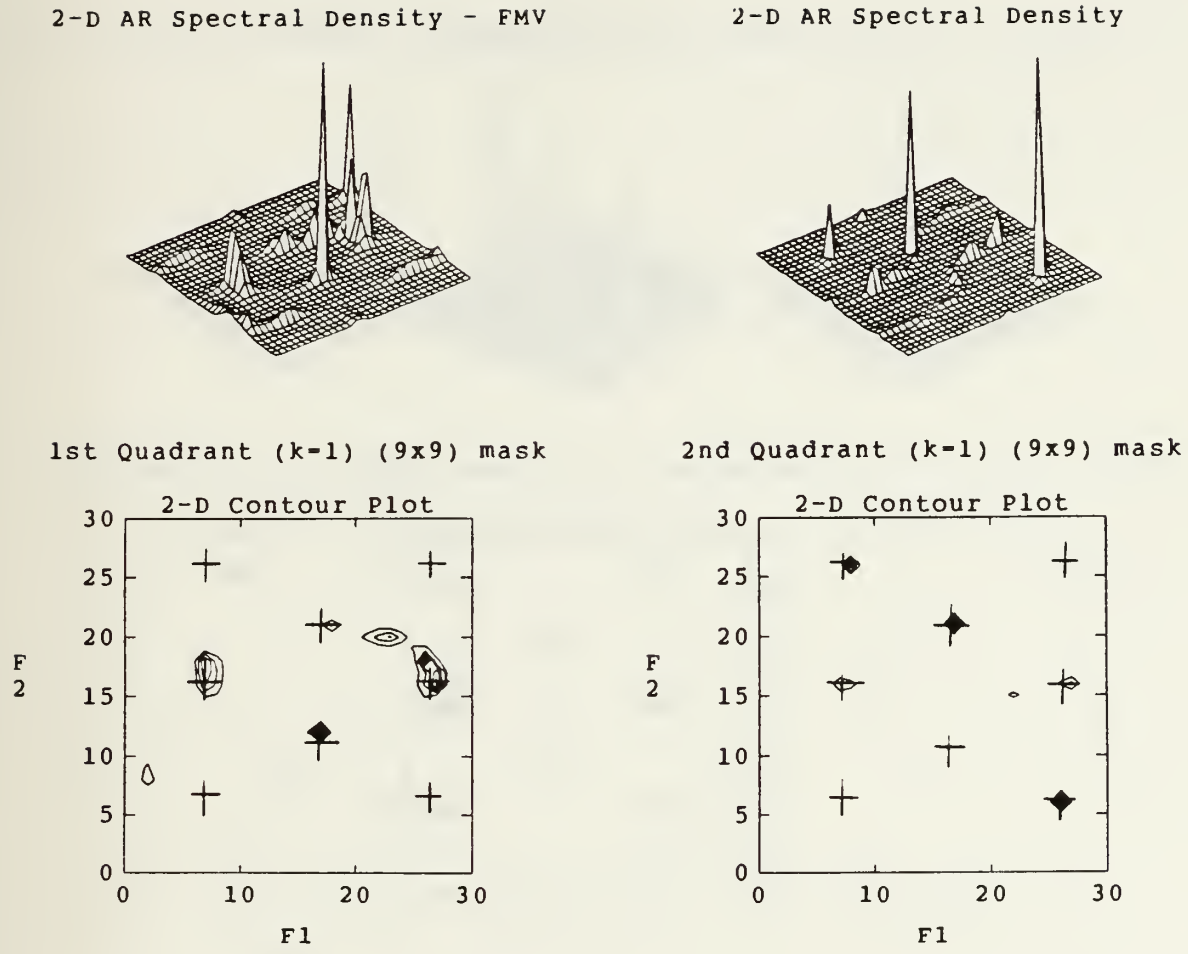

Figure D.5: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the *FMV-cycle* Method. 1st & 2nd quadrant estimates shown, (k=1, 9×9 mask).

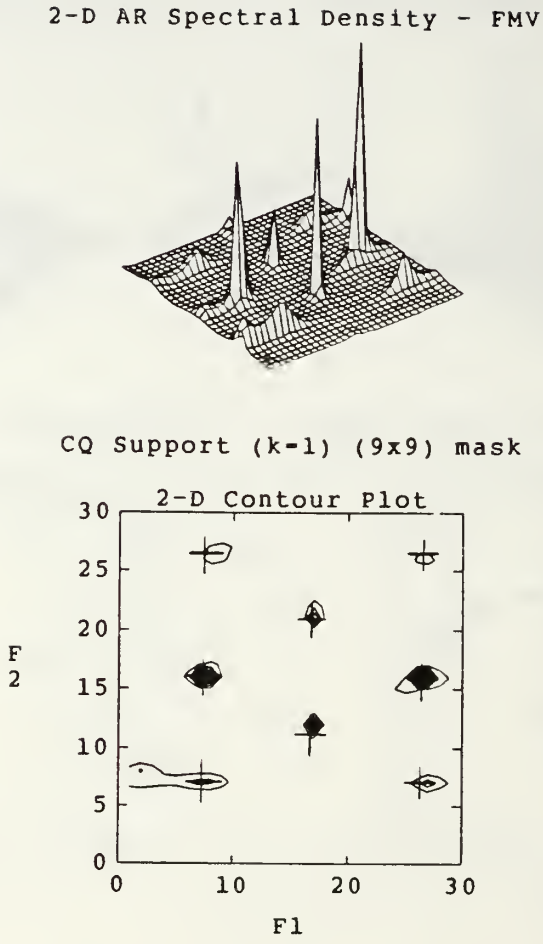

Figure D.6: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the *FMV-cycle* Method. CQ estimate shown, (k=1, 9×9 mask).

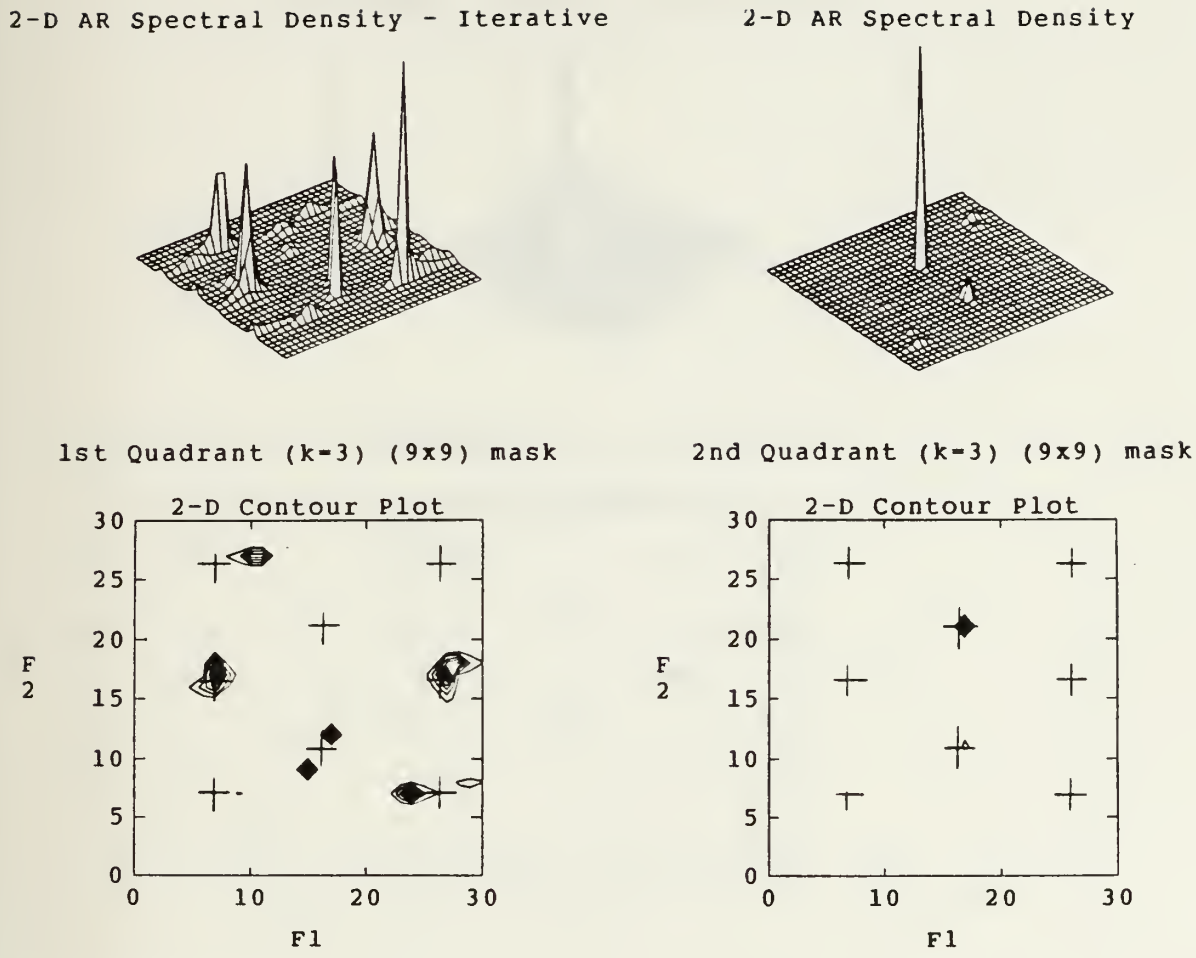

Figure D.7: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the Straight Iterative Method. 1st & 2nd quadrant estimates shown, (k=3, 9×9 mask).

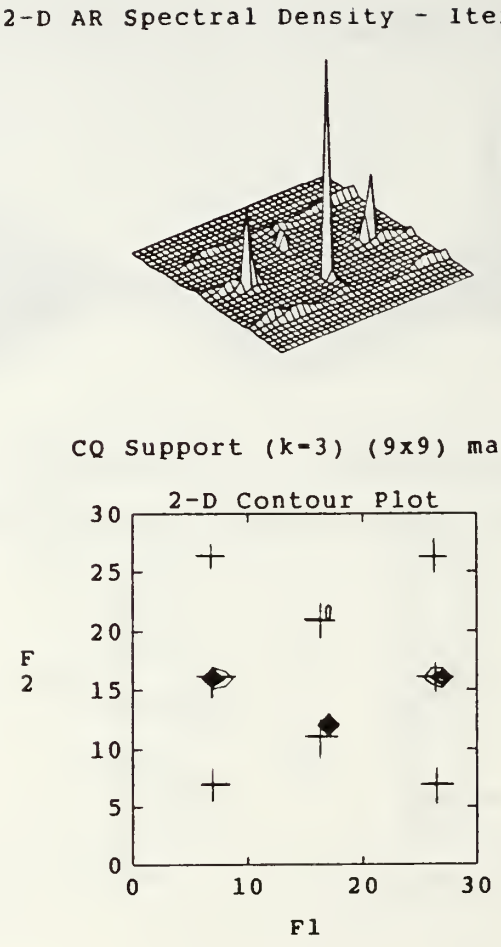

Figure D.8: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR=10dB) Using the Straight Iterative Method. CQ estimate shown, (k=3, 9×9 mask).

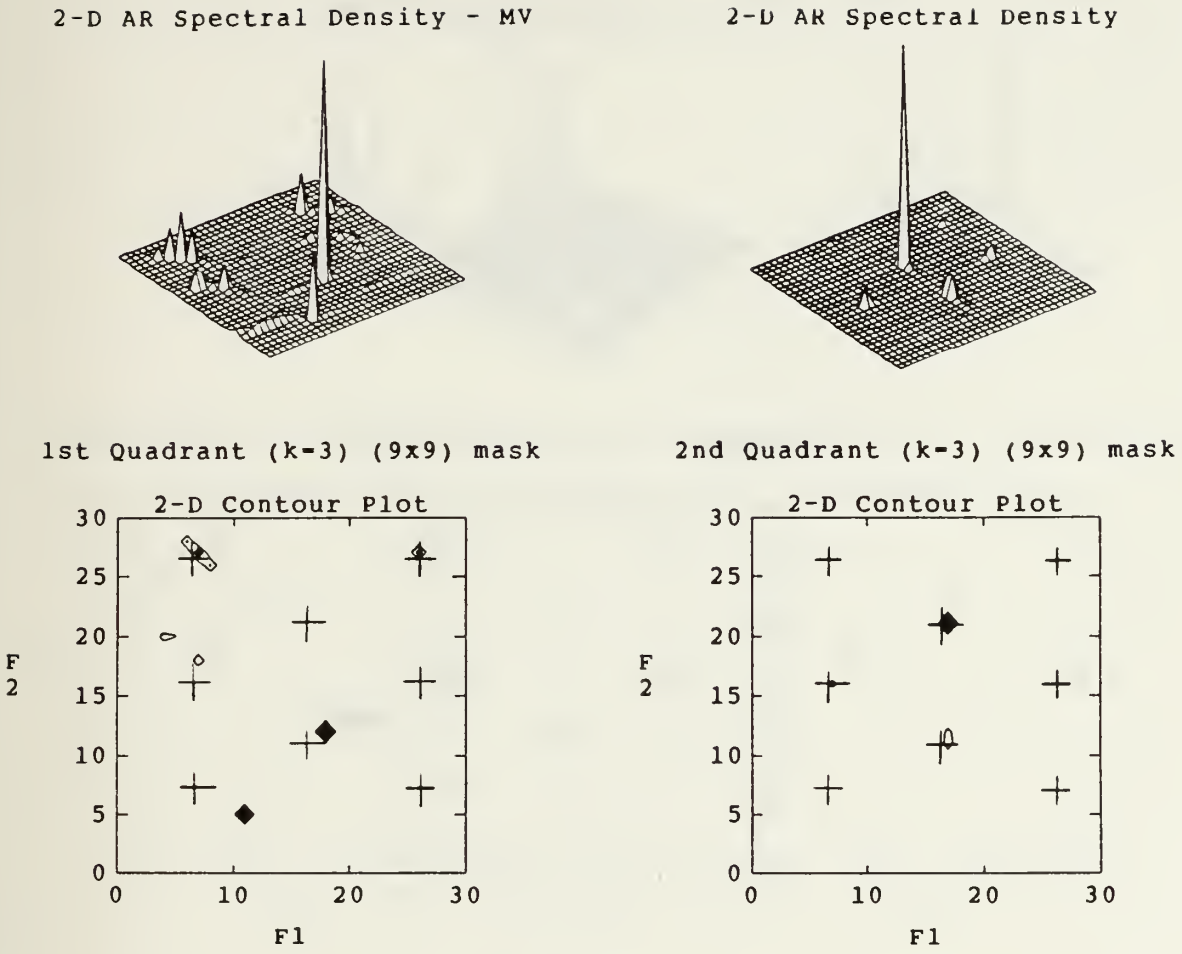

Figure D.9: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the *V-cycle* Method. 1st & 2nd quadrant estimates shown, (k=3, 9×9 mask).

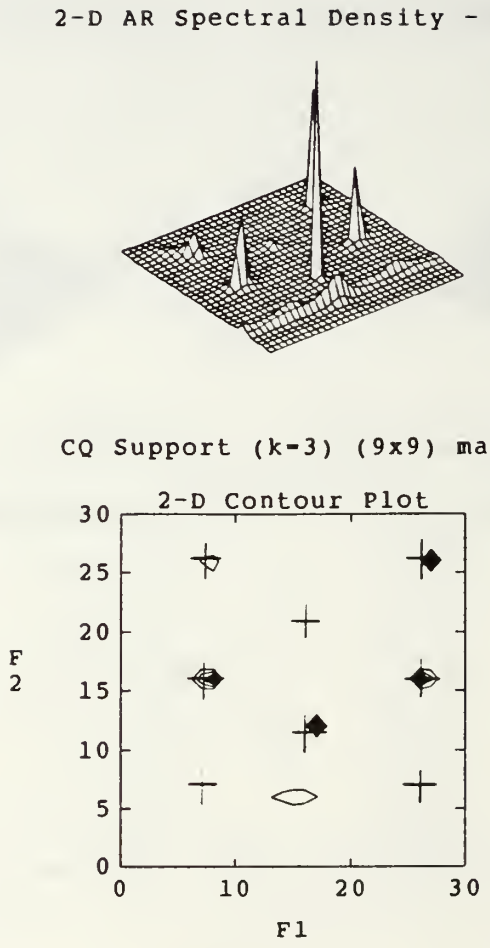

Figure D.10: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the *V-cycle* Method. CQ estimate shown, (k=3, 9×9 mask).

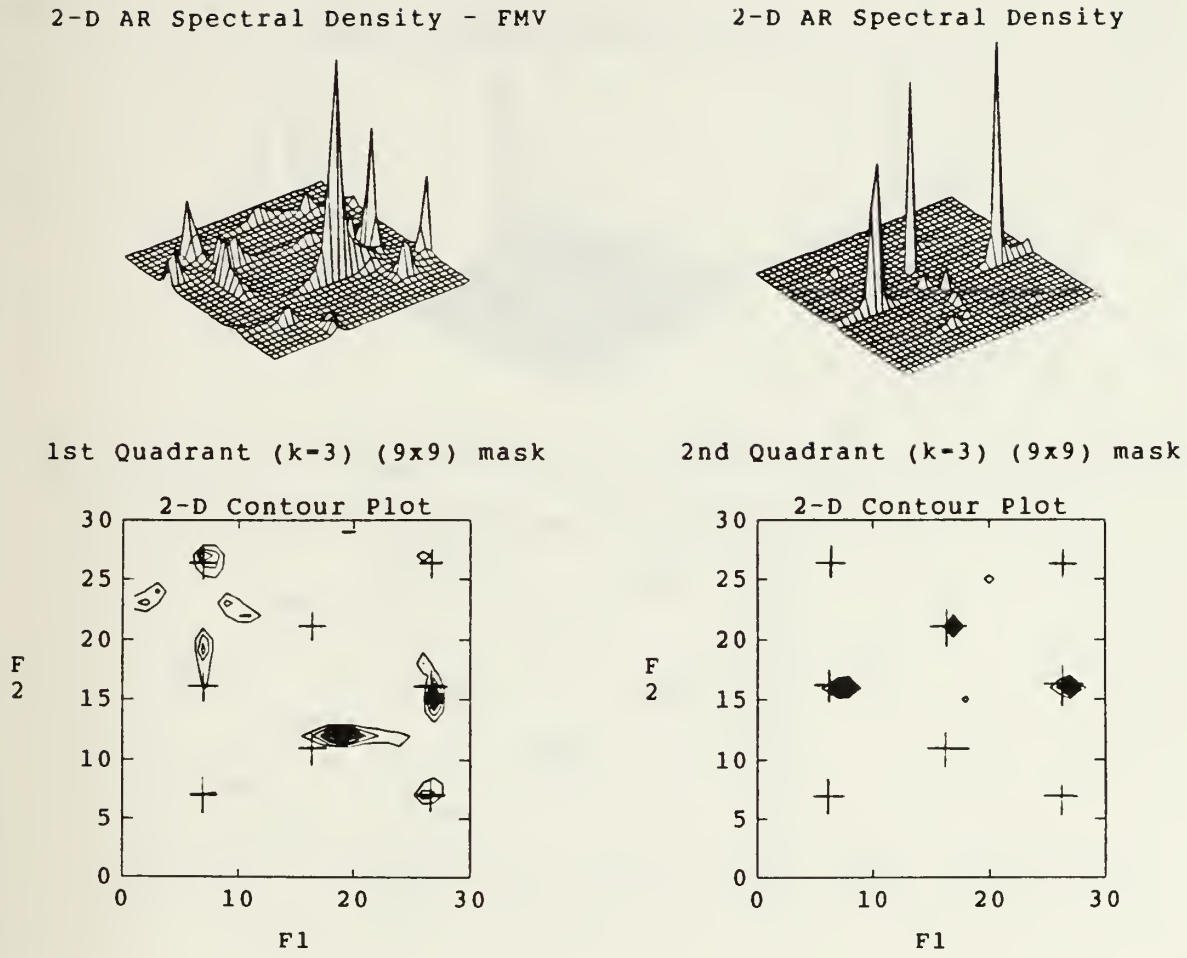

Figure D.11: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR=10dB) Using the *FMV-cycle* Method. 1st & 2nd quadrant estimates shown, (k=3, 9×9 mask).

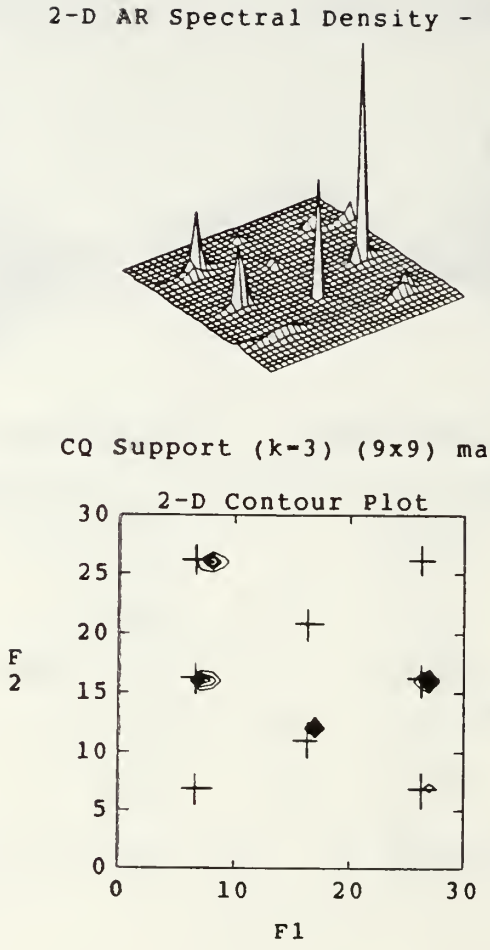

Figure D.12: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the *FMV-cycle* Method. CQ estimate shown, (k=3, 9×9 mask).

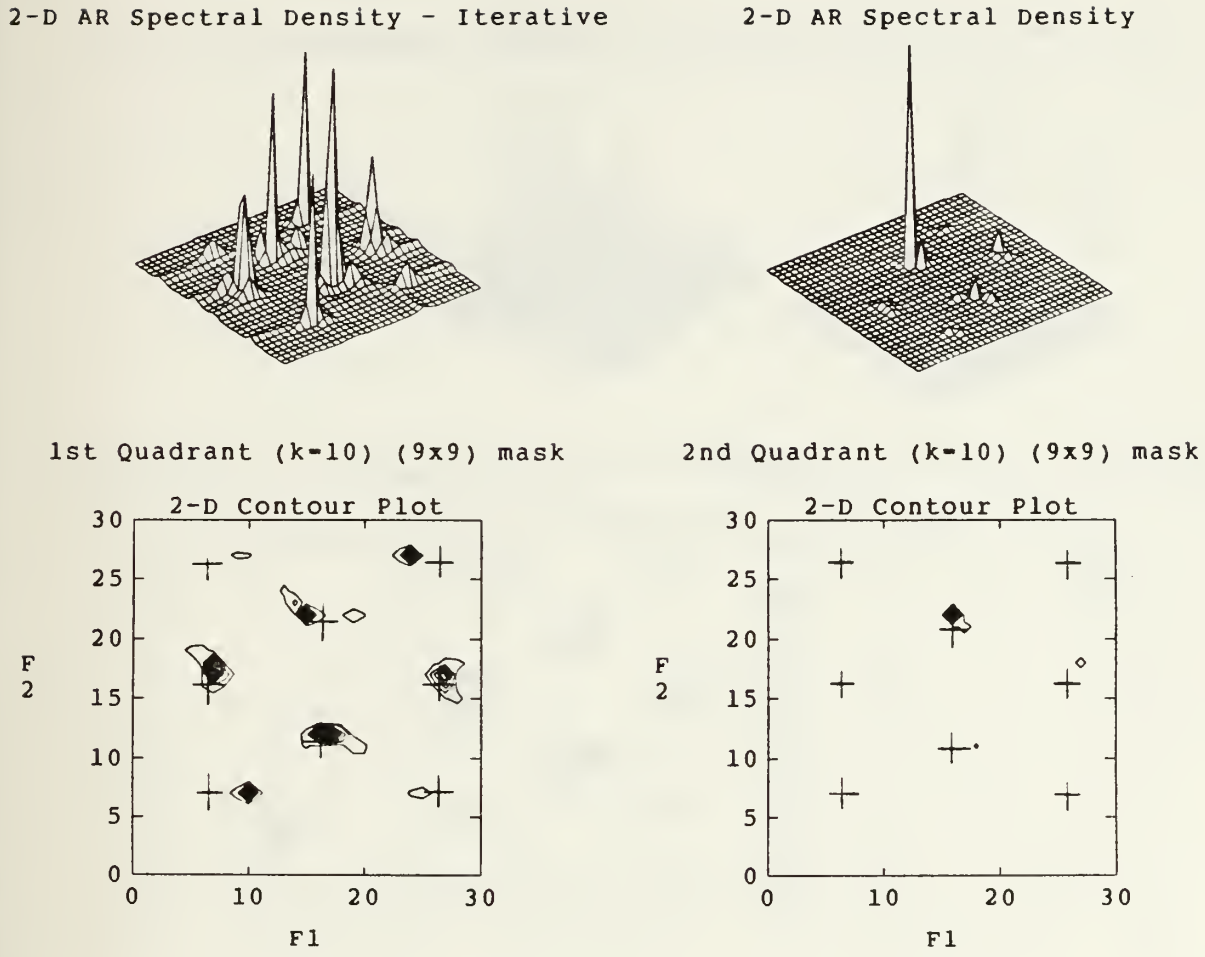

Figure D.13: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the Straight Iterative Method. 1st & 2nd quadrant estimates shown, (k = 10, 9×9 mask).

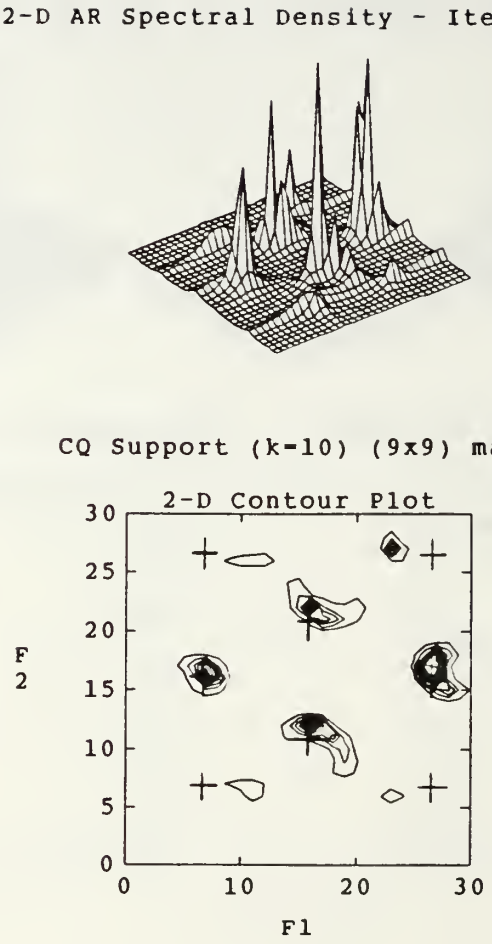

Figure D.14: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the Straight Iterative Method. CQ estimate shown, (k=10, 9×9 mask).

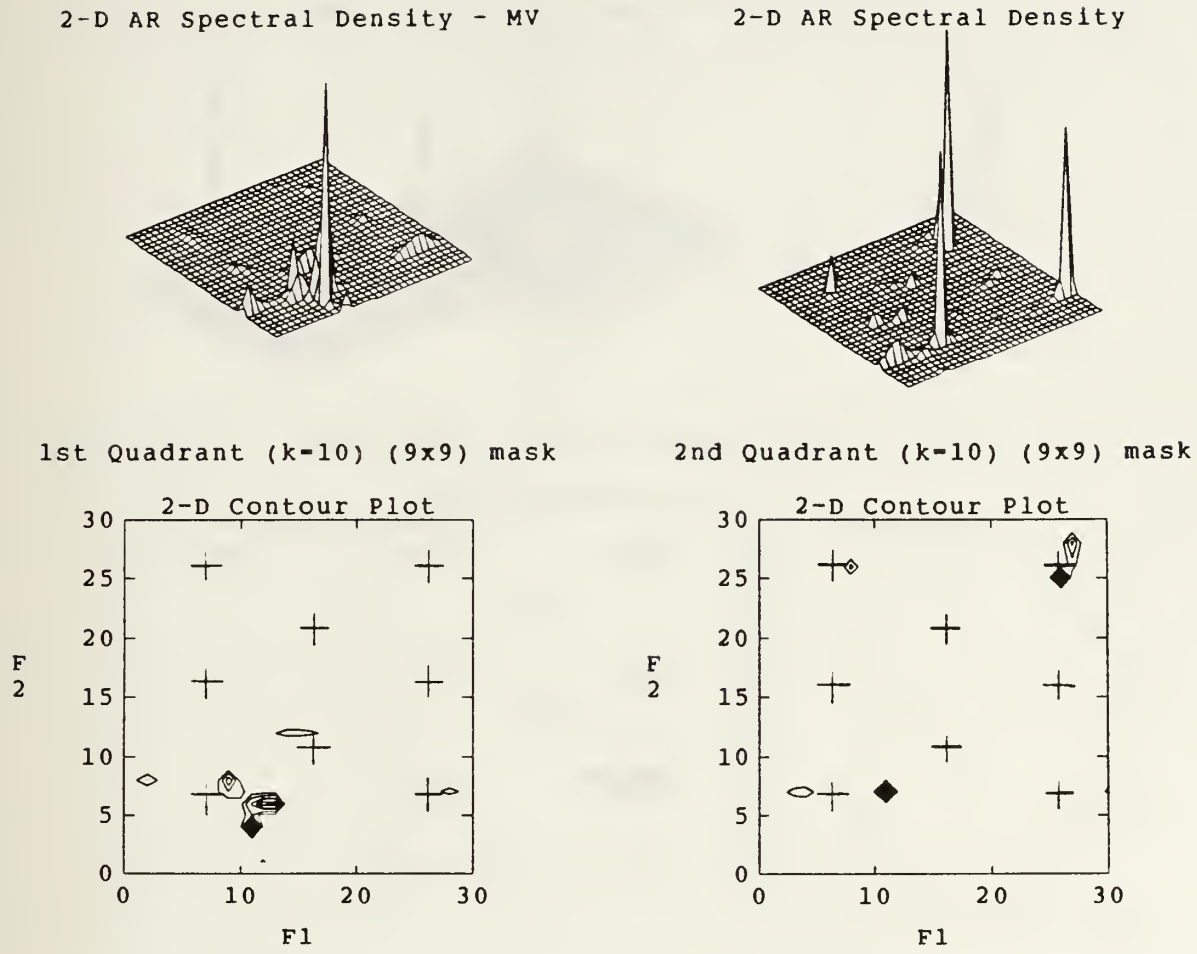

Figure D.15: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the *V-cycle* Method. 1st & 2nd quadrant estimates shown, (k=10, 9×9 mask).

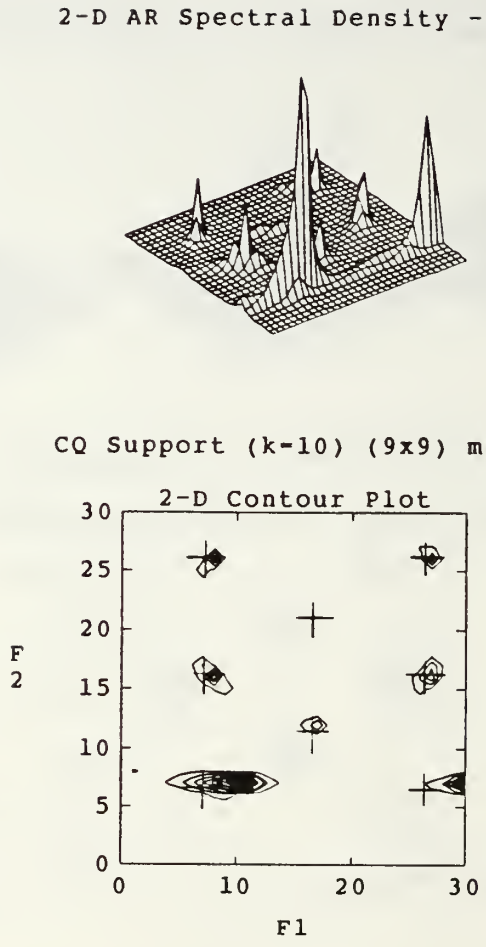

Figure D.16: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the *V-cycle* Method. CQ estimate shown, (k=10, 9×9 mask).

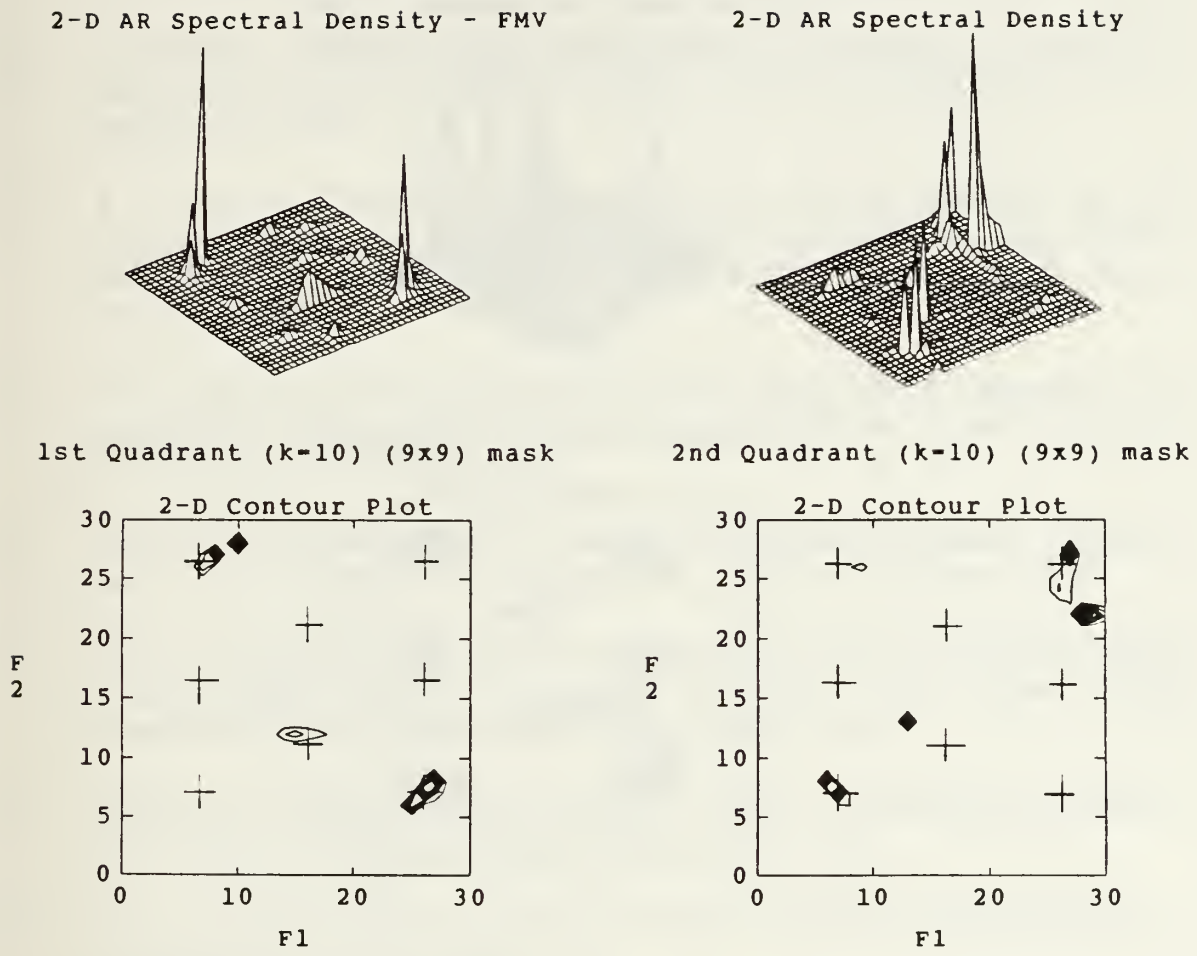

Figure D.17: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the *FMV-cycle* Method. 1st & 2nd quadrant estimates shown, (k=10, 9×9 mask).

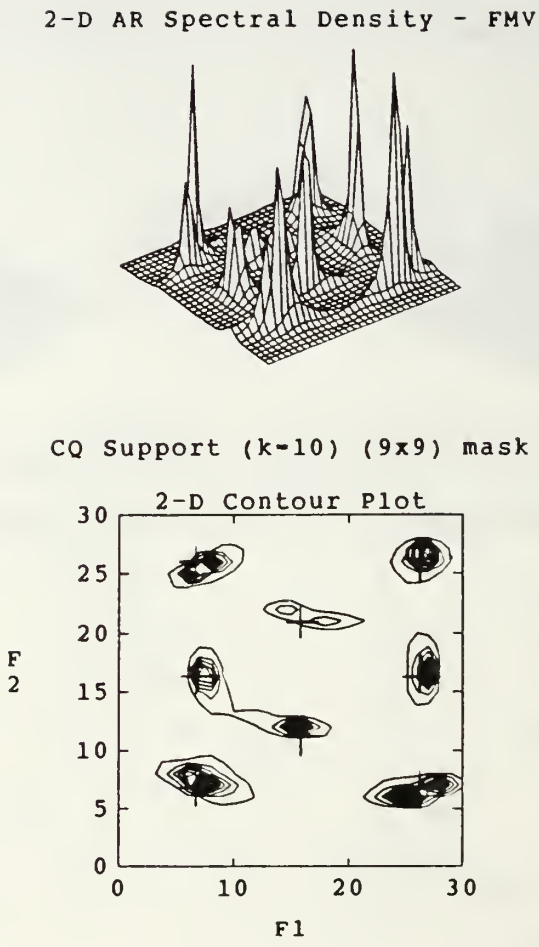

Figure D.18: 2-D AR Spectral Estimate of Eight Sinusoids in Noise (SNR = 10dB) Using the *FMV-cycle* Method. CQ estimate shown, (k=10, 9×9 mask).

REFERENCES

1. Charles W. Therrien, *Discrete Random Signals and Statistical Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey (to be published in 1992).

2. M. Tummala, "Efficient Iterative Methods for FIR Least-Squares Identification", *IEEE Trans. Acoustics, Speech, Signal Processing*, Vol. ASSP-38, No. 5, pp. 887-890, May 1990.

3. R. Wester, M. Tummala, and C.W. Therrien, "Iterative Algorithms for 2-D Spectral Estimation," in *Proc. Twenty Fourth Asilomar Conf. on Signals, Systems and Computers*, Pacific Grove, California, IEEE Computer Society Press, November 1990.

4. M. Tummala, "New Algorithm For Solving Block Matrix Equations with Applications in 2-D AR Spectral Estimation", *IEEE Trans. Signal Processing*, Vol. 39, pp. 759-764, March 1991.

5. William R. MacHardy, "Iterative Methods for Parameter Estimation", Master's thesis, Naval Postgraduate School, Monterey, California, December 1990.

6. John C. Eremic, "Iterative Methods for Estimation of 2-D AR Parameters Using a Data-Adaptive Toeplitz Approximation Algorithm", Master's thesis, Naval Postgraduate School, Monterey, California, September 1991.

7. William L. Briggs, *A Multigrid Tutorial*, Society for Industrial and Applied Mathematics, Philadelphia, Pennsylvania, pp. 7-54, 1987.

8. Jae S. Lim, *Two-Dimensional Signal and Image Processing*, Prentice Hall Inc., Englewood Cliffs, New Jersey, 1990.

9. Douglas O'Shaughnessy, *Speech Communication*, Addison-Wesley Publishing Co., Reading, Massachusetts, pp. 336-378, 1987.

10. Steven M. Kay, *Modern Spectral Estimation*, Prentice-Hall, Englewood Cliffs, New Jersey, pp. 31-33, 1988.

11. Simon Haykin, *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, New Jersey, pp. 370-401, 1991.

12. Alan V. Oppenheim and Ronald W. Schafer, *Discrete-Time Signal Processing*, Prentice-Hall, Englewood Cliffs, New Jersey, pp. 786-795, 1989.

13. L.B. Jackson and H.C. Chien, "Frequency and Bearing Estimation by Two-Dimensional Linear Prediction", *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, Washington, D.C., pp. 665-668, April 1979.

14. C.W. Therrien and H.T. El-Shaer, "A Direct Algorithm for Computing 2-D AR Power Spectrum Estimates", *IEEE Trans. Acoustics, Speech, Signal Processing*, Vol. 37, No. 11, pp. 1795-1798, November 1989.

INITIAL DISTRIBUTION LIST

	No. of Copies
1. Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2. Library, Code 52 Naval Postgraduate School Monterey, California 93943-5002	2
3. Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
4. Professor Murali Tummala, Code EC/Tu Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	4
5. Professor Charles W. Therrien, Code EC/Ti Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, California 93943-5000	1
6. Dr. R. Madan (Code 1114SE) Office of Naval Research 800 North Quincy Street Arlington, Virginia 22217-5000	1
7. Mr. John Hager (Code 70E1) Naval Undersea Warfare Engineering Station Keyport, Washington 98345	1

8. Commanding Officer 2
Attn: LT Dean A. Richter, USN
Naval Submarine School
Code 80 SOAC Class 92030
Box 700
Groton, Connecticut 06349-5700

Thesis
R3956 Richter
c.1 Iterative system
modeling using multi-
grid techniques.